Filter Your Email On OS X Server - With Open Source Tools!

MacTech Magazine
November • 2005

MACTECH®

The Journal of Macintosh Technology

Bonjour From The Source Hound - Automatic Service Discovery in Tiger

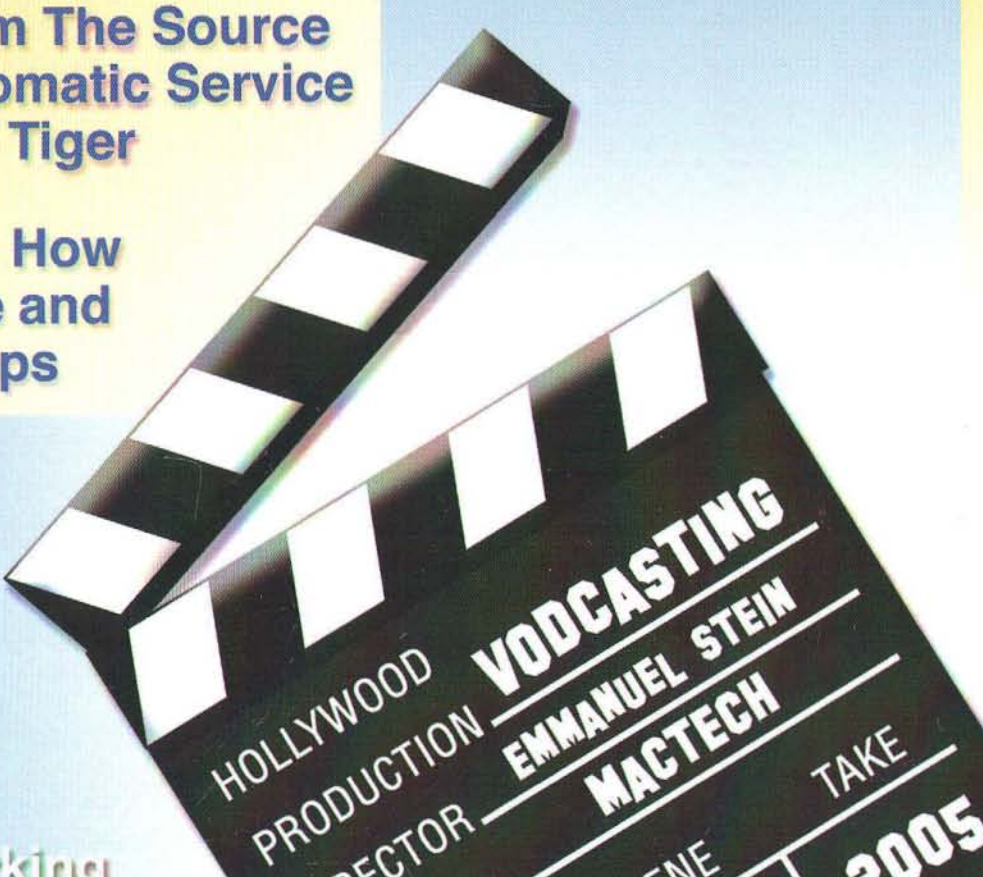
X11 & OS X - How To Configure and Run Your Apps

Mac In The Shell - Back To bash Basics All Over Again!

QuickTime Toolkit - Working with Properties

AppleScript Essentials - Intro to Scripting iCal

Fetch, Clarus, Fetch - Core Data Part III



Exploring the Future of Syndicated Media

New Featured Enterprise IT Column!

In The Trenches with Mike Harvey and Schoun Regan

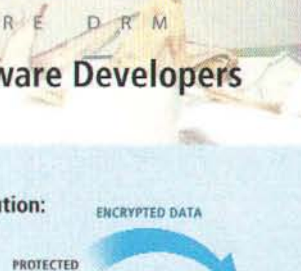
$8.95 US,
$12.95 Canada
ISSN 1067-8360
Printed in U.S.A.

3,248 hours typing code
184 hours finding that one bug
142 hours of meetings
108 pizzas
14 cancelled weekend trips
11 all-nighters
1 call protects it all!
HASP®
SOFTWARE DRM
#1 Choice of Software Developers
5:00 DEADLINE!

The new HASP family of products is the next generation in protection ensuring the highest level of security for your software. It provides an easy set of tools to automatically protect your software and implement new and innovative licensing options.

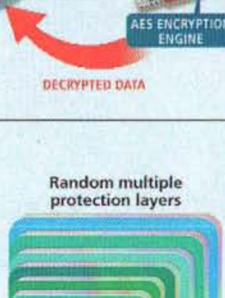
▶ Strongest anti-piracy solution:

Our powerful, 128-bit AES encryption provides a strong locking mechanism that ensures you get paid for every copy of your software.



▶ Robust envelope:

Automatically protects Windows executables, DLLs, .Net and Mac applications, securing the weakest link and ensuring your valuable IP remains unreachable. Multiple protection layers provide unequaled security.



▶ Best overall solution as rated by an independent testing lab:

In KeyLabs tests, HASP HL outperformed the competition in security, ease-of-use, flexibility of tools and functionality, and compatibility across platforms and environments. A copy of the full KeyLabs report is available from our Web site at www.aladdin.com/hasp.

Category	Weight	Aladdin HASP HL	Sentinel UltraPro
Security	35%	85.1%	60.9%
Ease-of-use	25%	93.7%	77%
Licensing	20%	75%	75%
Flexibility of tools and functionality	10%	93.3%	71.2%
Compatibility across platforms and environments	10%	93.3%	71%
Overall Evaluation		87.3%	69.8%

Request a Software Developer Kit today at www.aladdin.com/hasp

Aladdin®
SECURING THE GLOBAL VILLAGE

North America: 1-800-562-2543, 847-818-3800 • UK • Germany • Israel • Benelux • France • Spain • Asia Pacific • Japan

Need to find something fast?

INDEX
INDEX
INDEX

With c-tree Speed.

TRY IT NOW!

FairCom's c-tree Plus®

embedded database engine offers Superior Indexing Technology – the key to performance, data integrity, and concurrency. c-tree Plus offers direct record-oriented C and C++ APIs with an industry-standard SQL interface that allows use of any combination of APIs within the same application. Furthermore, we offer source code access for intimate programming control, unmatched portability, and developer-to-developer technical support.

Heterogeneous Environments

c-tree Plus and c-treeSQL™ Servers are the perfect solution for your mixed platform environments. Mac servers to Windows clients? No problem. Linux Servers to Mac clients? No problem. c-tree has a long history of cross-platform development solutions. Byte incompatibility between platforms is handled seamlessly with our Uniformat data handling technology.

Low TCO

c-tree is priced affordably, requires minimal hardware resources, and needs no IT staff for maintenance. If Total Cost of Ownership (TCO) is important to you, c-tree is the perfect database.

Easy Deployment

c-tree Servers are designed for ease of use and deployment as well. Out of the box, our servers can be installed and running in minutes.

Start indexing your data today!

FairCom®

USA • Europe • Japan • Brazil

www.faircom.com

Go to www.faircom.com/go/mteval for a FREE evaluation of c-tree Plus!

Data Banks Communications Inc

HOME OF THE FREE COMPUTER
THAT'S RIGHT FREE COMPUTER!

GET A MAC MINI OR DELL COMPUTER FREE JUST SIGN UP FOR VOIP DSL BUNDLE

INFOSPEED 768K/128K	$ 69.95
INFOSPEED 1.5M/128K	$ 69.95
INFOSPEED 1.5M/384K	$ 69.95
INFOSPEED 7.1M/768K	$ 199.95
INFOSPEED 384K/384K	$ 69.95
INFOSPEED 768K/768K	$ 149.95

TeleSoho 1.5M/128	$69.95
TeleSurfer Plus 608K/128K	$69.95
Residential 768K/128K	$69.95
SOHO1.5M/768K	$79.95
SOHO3.0M/384K	$79.95
SOHO3.0M/768K	$79.95
SOHO6.0M/768K	$139.95
TeleSpeed 144K/144K	$125.95
TeleSpeed 192K/192K	$125.95
TeleSpeed 384K/384K	$149.95
TeleSpeed 768K/768K	$159.95
TeleSpeed 1.1M/1.1M	$199.95
TeleSpeed 1.5M/1.5M	$249.95

Ask for business line special!

www.databanksglobal.com

For questions regarding sales or service, please call us 24 hours a day at 1-866-624-6114

MACTECH

A publication of
XPLAIN CORPORATION

The Editorial Staff

Publisher
Neil Ticktin
Associate Publisher
David Sobsey
Editor-at-Large
Dave Mark
Graphics & Production
Dennis Bower

Xplain Corporation Senior Staff

Chief Executive Officer
Neil Ticktin
President
Andrea J. Sniderman
Accounting
Marcie Moriarty
Customer Relations
Susan Pomrantz
Board of Advisors
Steven Geller
Alan Carsrud

Regular Columnists

QuickTime ToolKit
by Tim Monroe
Reviews/KoolTools
by Michael R. Harvey
Patch Panel
by John C. Welch
AppleScript Essentials
by Ben Waldie
The Source Hound
by Dean Shavit
Mac In The Shell
by Ed Marczak

Board of Advisors

Dave Mark, **Chairman**

Jordan Dea-Mattson, Steven Geller, Bruce Friedman, and Richard Kimes

Contributing Editors

Michael Brian Bentley
Vicki Brown
Marshall Clow
John C. Daub
Tom Djajadiningrat
Andrew S. Downs
Gordon Garb
Chris Kilbourn
Rich Morin
Will Porter
Avi Rappoport
Cal Simone
Steve Sisak

TABLE OF CONTENTS

FEATURED ARTICLES

MacTech Magazine (ISSN: 1067-8360 / USPS: 010-227) is published monthly by Xplain Corporation, 850-P Hampshire Road, Westlake Village, CA 91361-2800. Voice: 805/494-9797, FAX: 805/494-9798. Domestic subscription rates are $47.00 per year. Canadian subscriptions are $59.00 per year. All other international subscriptions are $97.00 per year. Domestic source code disk subscriptions are $77 per year. All international disk subscriptions are $97.00 a year. Please remit in U.S. funds only. Periodical postage is paid at Thousand Oaks, CA and at additional mailing office.

POSTMASTER: Send address changes to **MacTech Magazine**, P.O. Box 5200, Westlake Village, CA 91359-5200.

Automate your media workflow on your desktop and server.

Now you can automate hundreds of repetitive media editing, optimization, conversion and delivery actions on the desktop and server with a simple drag and droplet.

FTP
ZIP/email
web
mobile
dvd/cd
print
client HD
server HD

Introducing the next generation in media automation—the new "action packed" Equilibrium® DeBabelizer® Pro 6 Client & DeBabelizer Server for Mac® OS X/Xserve.

Deliver high-quality, high-speed results, while saving you time and money in the process. Automatically prepare your images, animations and digital video files on your desktop for optimal multi-channel delivery with the new Equilibrium DeBabelizer Pro 6. When you purchase the new DeBabelizer Server (free trial included), you and your team can send numerous media processing jobs to the server level for lightning fast results delivered to any off-line location, FTP site, and back to you, or Zip and email to anyone on demand. Whether you're designing content for websites, mobile devices, DVDs, video games, interactive applications and print, Equilibrium DeBabelizer Pro 6 and DeBabelizer Server kick assets.

Call 866.EQUILIB or visit www.equilibrium.com to upgrade now to DeBabelizer Pro 6 for only $199.95

Watch our demonstration video and enter to win a free iPod now: www.equilibrium.com/win_iPod

equilibrium®
Automated Media Processing Solutions

Includes 50 free DeBabelizer Pro 6 Automator Actions for "Tiger" and create your own with a single click!

Mac

MACTECH

Communicate With Us

Department E-Mails

Orders, Circulation, & Customer Service
cust_service@mactech.com

Press Releases
press_releases@mactech.com

Ad Sales
adsales@mactech.com

Editorial
editorial@mactech.com

Online Support
online@mactech.com

Accounting
accounting@mactech.com

Marketing
marketing@mactech.com

General
info@mactech.com

Web Site
http://www.mactech.com

In this electronic age, the art of communication has become both easier and more complicated. Is it any surprise that we prefer **e-mail?**

If you have any questions, feel free to call us at 805/494-9797 or fax us at 805/494-9798.

If you would like a subscription or need customer service, feel free to contact MacTech Magazine Customer Service at 877-MACTECH

We love to hear from you! Please feel free to contact us with any suggestions or questions at any time.

Write to letters@mactech.com or editorial@mactech.com as appropriate.

DEPARTMENTS

Talking With One Man on the Front Lines of Mac IT

First in a New Monthly Series!

Ever been to Macworld, WWDC, MacRetreats, or an Apple Certified Training course. If the answer is yes, then chances are you have met Schoun Regan. Author of the Mac OS X Server Visual Quick Pro guide from Peachpit Press, editor for most of the Apple Pro Training series books focusing on Mac OS X and Mac OS X Server, Schoun is one of the most powerful forces behind technical training and Mac OS X in the industry today. The list of people he calls his colleagues is a veritable who's who of the Mac OS X high end consulting community. His ideas about the merging of training and consulting surrounding Mac OS X have been accepted by many who listen to him speak, hire his company to develop courseware, or have them consult on deployments of Mac OS X and Mac OS X Server. Arguably, he has been associated with training and Apple operating system software for about as long as the Macintosh has been around. We recently had the chance to sit down with Schoun and get his take on the current state, and future direction, of Macintosh in the enterprise.

MacTech: Where did you get your start in the IT business, and how long have you been at it?

Schoun Regan: I started out at Goodyear Research, where they were testing the strength and elasticity of tire materials. We were able to play with all sorts of dangerous equipment, data collection cards, IBM-XTs, Token Ring networks, AppleTalk, and Novell. It was a blast. After ten years supporting scientists, chemists, and engineers there, I moved on, earned one of the first ACT certifications, and established The Mac Trainers, and later ITInstruction.com. I have taught, and supported Mac OS X/Server installations all over the country, and speak at many Macintosh related events throughout the year.

MT: Apple has been making serious inroads into the enterprise computing space over the last few years. How do you think they got to where they are now?

SR: Hard work and innovation. The engineers started understanding the importance of integration with other platforms. They are starting to listen more, and great things are happening because of that. Take Access Control Lists in Tiger, for example. The big things get the headlines, but it's the little things that administrators deal with every day that change their minds. Take a look at http://www.apple.com/server/documentation. Over 1200 pages of detailed documentation on Mac OS X Server, FREE. The Mac OS X Server team is headed up by some very forward thinking people, and it shows. Apple Certified Training courses also help to spread the news that Apple is a player in this market.

MT: What were some of the challenges they had to overcome to get where they are today?

SR: Myopic mindsets with regard to Mac OS X Server had to go, and it did very quickly. Another issue they had to change was customer impressions. For a Fortune 500 company administrator to say, "We'll never have Macs in our company", is indicative of the larger problem of insular thinking. Apple's sales force is working diligently to alter that perception. It's a tough road with plenty of closed doors displaying "No Solicitors" signs. But those at Apple persevere. I recently spoke to a person who left Sun computing and they remarked his Apple SE seemed genuinely happy to go above and beyond the call of duty. There are not so much technical challenges as personnel challenges. The team heading up Sales, especially for Enterprise, is outstanding. I would attribute a significant change in customer mentality towards Mac OS X to these people.

MT: So how do companies make the switch, on an Enterprise level, so to speak?

SR: It's not a simple process. Getting the buy-in of the department manager or CIO is a great start. You have a section of the administrative world that still sees Macs plus AppleTalk multiplied by messy networks equals "we don't understand them". I once told the CIO of a company, when speaking about Apple entering the larger business markets, "Apple does not deserve my loyalty, but they have earned my respect." Someone needs to be in front of these administrators and let them know that Mac OS X Server is here to stay. It's interesting to see their response when shown what Mac OS X and Mac OS X Server can do. The market is ripe for a change, and Apple knows this. You have to more hardware and more staff ensures job security. Mac OS X Servers are a great start. Laptops and Mac OS X desktops can come later. Selling the hardware is only part of the solution. For someone to change over, a long-term plan must be in place meeting the criteria I spoke of earlier.

MT: What do you see as the bumps on the road to getting there?

SR: I would guess that the percentage of sales of Xserve and Xserve RAIDs cannot come close to iPod sales. However, you have the opportunity to grow other aspects of the business off sales of Mac OS X Server related products; training, repair kits, extended warranties, solutions, etc. When servers go in, Mac OS X soon follows. Ergo, listening to customers about issues surrounding Mac OS X Server should be taken seriously. Security is an enormous factor. Apple has fantastic security architecture. Do we see that in Apple's marketing literature or on the web site? Apple has an Active Directory plug-in allowing Mac OS X to integrate seamlessly with an Active Directory domain. Why this is not promoted better in Windows based PC magazines and web sites is beyond me.

MT: So, let's say you are king of the world; what would you have Apple do to improve their position in large scale IT departments?

SR: Fully support the extended schema by both offering phone support and training to allow easier integration of Windows clients. Packet signing and other similar services that Active Directory offers to its clients. Spend the money and purchase Thursby Software outright. I know that's a Microsoftee thing to do, but they have exactly what Apple needs, and they'll gain some talent found nowhere else in the industry. Develop WWTC, a World Wide Technical Conference. This can encompass WWDC, the World Wide Developer Conference and WWIT, a World Wide Information Technology Conference. I'd love to head that up. With the move to Intel, both aspects of this transition should be explored. This conference would take the IT world by storm.

MT: What is your perception of Apples move to the Intel processor? Good move or bad?

SR: Heh. Buy Apple stock.

MT: Thanks for taking the time do talk with us.

SR: My pleasure. I think this type of forum is productive. There are many Macintosh IT professionals out there, and I know many of them. You should hear some of the things they have to say.

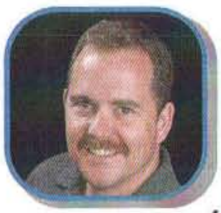
MT: Cool. Want a job?

SR: *(laughs)*

Editor's note: Shortly after this interview, we actually did offer Schoun the job. Beginning with the December issue of MacTech, each month Schoun will be interviewing a Mac IT manager, and sharing with you their strategies for dealing with their most pressing issues. Be looking for "In The Trenches" each month in MacTech!!

MT

About The Author



"Honest to God, no s#@&, there I was." Thus begins the story of Reviews Editor Michael R. Harvey, a man who only wishes he could have slept his way to the top. From his humble beginnings as a helicopter crew chief in the U.S. Army, serving in both the Panama invasion and the first Gulf War, to UC Santa Barbara, earning a B.S. in Aquatic Biology, he gained his Mac experience in various places. From his early super hero days as the computer go to geek in freshman dorm, on to several positions in the undergrad labs on campus and with various consulting firms, to his current secret identity as the Senior Systems Technician at the Ventura County Star newspaper. Add to that other early jobs as a pizza delivery boy, and bouncer, and Michael was perfectly poised to take the Reviews Editor job in July 2002, wrangling both writers and vendors into line to be able to bring to you reviews of cool, and yes, even useful, products (even penning a few choice pieces himself). You're welcome. Contact him at reviews@mactech.com

Bonjour, Mon ami: Automatic Service Discovery in Tiger

When Apple released Mac OS X 10.2 in August of 2002, it included a somewhat obscure networking technology dubbed Rendezvous. Nearly than three years later with the release of OS X 10.4 Tiger, Mac users have said ***au revoir*** to Rendezvous and hello to Bonjour because of a trademark infringement lawsuit against Apple by Tibco Software. The new name for Rendezvous was to be OpenTalk, which had a familiar, warm Classic Mac OS ring, but instead, we walk around with a frog in our throat, lips shaped like the Texas border, trying to emulate an uncomfortable accent. Every time I say the B-word I have visions of Pepe LePew sticking his head out of a doorway and exclaiming, "Bonjour, mon amour, embrasse-moi." Freaky.

Yet the renewed focus on this taken-for-granted technology, no matter how odd the name or the freely associated cartoon character, reveals that it has become such a useful and ubiquitous helpmate that it's not difficult to make the argument that without Bonjour, OS X would be a lesser experience for everyone from the casual iPod user to the Education Systems Administrator with dozens of XServes and XServe RAIDs to manage. For example, the requisite RS-232 serial connection required to configure nearly every storage subsystem on the market from a SCSI array, NAS or high-end switch, is easily dispensed with in favor of Bonjour. Just plug a PowerBook into the Ethernet port on the back of the XServe RAID, and in seconds they're talking, all with no manual IP configuration, and no serial port. While web-based configuration is the norm for such devices, a trip to the serial port is almost always required to configure the IP address of the web interface first.

Zero, My Hero, How Wonderful You Are...

By the subhead above, I'm probably giving away my age. I grew up with the Schoolhouse Rock cartoons on Saturday morning television. I was especially a fan of Multiplication Rock, where the multiplication tables were brought to life as cartoon characters. Zero, of course, was the ultra-powerful multiplier, portrayed as a superhero, complete with mask and cape. If you've been reading this column for a while, you'll know that while I might appear to have cartoons on the brain, there's usually a point, at least an intersection of Bullwinkleish meaning with something Open-Source. At the core of Bonjour is an Open Source (surprised?) technology known as Zeroconf (http://www.zeroconf.org).

One of the Classic Mac OS's claims to fame (and undoing) was the ease of use and automatic discovery of network shares and printers via the AppleTalk networking protocol. While AppleTalk was excellent for small groups of Macs, running it on larger networks and over WAN connections required special planning, hardware, and "seed routers." Contemporaries of AppleTalk, NetBIOS and Novell's IPX (Internetwork Packet Exchange) also provided facilities for discovery of network resources via broadcast. In the end, it was that broadcasting and lack of compatibility with TCP/IP, which became the darling standard of corporate networks in the mid-1990s, not just the Internet, that doomed AppleTalk to a deprecated protocol. Now, with Tiger, file sharing over AppleTalk isn't just deprecated; it doesn't work at all. When Leopard's released in late 2006 or early 2007, I wouldn't be surprised if AppleTalk wasn't supported for printing as well as file sharing.

The Zeroconf project outlines the following requirements in achieving what it calls the "AppleTalk ease-of-use in IP":

- Allocate addresses without a DHCP server (IPv4 Link-Local Addressing)
- Translate between names and IP addresses without a DNS server (Multicast DNS)
- Find services, like printers, without a directory server (DNS Service Discovery)
- Allocate IP Multicast addresses without a MADCAP server (future work)

A final requirement is that the solutions in the four areas must coexist gracefully with larger configured networks. Zeroconf protocols MUST NOT cause harm to the network when a machine is plugged into a large network.

It is important to understand that the purpose of Zero Configuration Networking is not solely to make current personal computer networking easier to use, though this is certainly a useful benefit. The long-term goal of Zero Configuration Networking is to enable the creation of entirely new kinds of networked products, products that today would simply not be commercially viable because of the inconvenience and support costs involved in setting up, configuring, and maintaining a network to allow them to operate.

The idea here is to allow machines of disparate operating system to easily "find" each other without needing to configure a network interface. The example given at the Zeroconf site is of two people wanting to play a networked computer game. If both are using PowerBooks, it ought to be a cinch to use AppleTalk for the two opponents to find each other, but if one's using a Windows laptop, then TCP/IP would be the only common network protocol (At least prior to OS X), and in the absence of a DHCP server for the players' home network, manual IP configuration would be a requirement for game play, and that's something many casual computer users still find somewhat difficult, and a process that game publishers really can't afford to support with their in-house staff. Part of the solution has been to use "tracker" servers to register game players, but that doesn't really address the issue of an ad-hoc network of two gamers who might not have an Internet connection at their disposal.

It's pretty much taken for granted now that physically connected computers (this includes those on wireless—Airport—networks as well) can discover each other's services. The Sony PSP (PlayStation Portable) features ad-hoc 802.11b support, the same as Apple's original Airport implementation that allows players running the same game to "find" each other when in range.

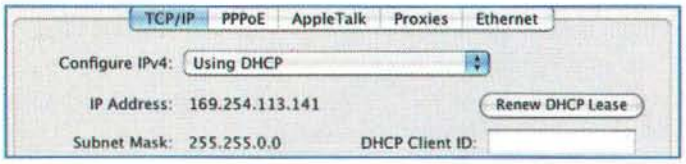
Life Without a Serial Port

"Entirely new kinds of networked products. . ." is exactly the promise that the XServe RAID fulfills in allowing itself to communicate without an assigned IP address over an Ethernet connection, as well as an XServe in headless setup mode. Airport base stations, iTunes music libraries, iPhoto libraries and more ***just show up*** to those looking for such services. Not surprisingly, the very first devices to take advantage of Zeroconf capabilities were network printers, which are often a network admin's worst nightmare when it comes to assigning static IP addresses. There's nothing more frustrating than standing over a hard-to-read LCD screen tapping on little buttons that require you to cycle from zero to 255 just because you can't ***go down as well as up***. While printers have gotten better in that department (HP, that means you), it's not unusual to run into printers left pulling IP addresses via DHCP because someone doesn't want to navigate the counter-intuitive menu tree on a tiny LCD screen.

Much of Bonjour's zero configuration magic lies in what's called "IPv4 Link-Local Addressing," which consists of IP addresses in the 169.254.x.x network range. Regardless of whether a Mac or another device on a network has either a manually assigned or DHCP assigned IP address, all OS X computers maintain a 169.254.x.x Link-Local address in their routing table in the event that a device with Link-Local addressing only shows up on the network, first polling other machines to make sure it gets a unique address. Using the `nestat -r` command to dump the routing table to standard output reveals the Link-Local network destination:

```
Internet:
Destination        Gateway            Flags    Refs      Use  Netif   Expire
default            192.168.0.1        UGSc       13        6    en0
127                localhost          UCS         0        0    lo0
localhost          localhost          UH         41   340679    lo0
169.254            link#4             UCS         1        0    en0
169.254.113.141    link#4             UHLW        1      889    en0
192.168.0          link#4             UCS         6        0    en0
192.168.0.1        0:0:94:83:34:68    UHLW       13        0    en0     1112
Dual2Ghz           localhost          UHS         0       19    lo0
192.168.0.255      link#4             UHLWb       1     6089    en0
```

Should the DHCP server on the network fail, it's likely that Mac OS X clients could continue printing and even mounting file server volumes via Bonjour. To see if a computer is using a Link-Local address instead of an assigned IP address, check the interface in the network preferences.



Configure IPv4: Using DHCP
IP Address: 169.254.113.141
Subnet Mask: 255.255.0.0
DHCP Client ID:

Figure 1. Link-Local IP Address

Even with this self-assigned Link-Local IP address (which is usually taken as a sign of trouble by those entrusted to maintain networks), it's still possible to do business as usual, provided that clients connect to the Bonjour name of the server, which is set to "Computername.local" by default.

Connecting To Server

Connecting to afp://dual2ghz.local ...

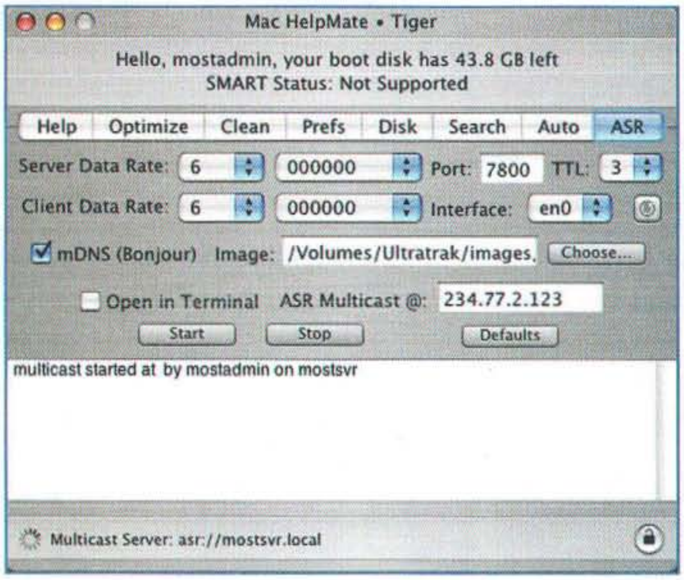
Figure 2. Connecting to an Apple File Server using Bonjour.

Many Windows 2000 and 2003 Server networks are mistakenly set up with an internal DNS domain ending in .local, which of course interferes with Bonjour service discovery. While it makes some sense to some (not to me) to use an internal domain name that's "not real," (mostly because people are of afraid of, or don't understand how DNS works) it might be a better idea to use .lan or .internal instead of .local. If it becomes apparent that the Bonjour .local extension is interfering with Windows Active Directory services. it's easy to permanently (but not irreversibly) disable Bonjour via the following `launchd` command:

```
launchctl unload -w
/System/Library/LaunchDaemons/com.apple.mDNSResponder.plist
```

mDNSResponder

While Bonjour might exist on most TCP/IP networks in the shadows of the routing table, or as a stepping-stone to help facilitate connections between Macs with DHCP or manually assigned IP addresses, it's also very useful in other situations, such as mass deployments, in conjunction with Apple's new multicast ASR (Apple Software Restore) capability. For example, when rolling out many new computers, it's sometimes desirable to simply plug them into an isolated switch, boot them from a custom (normally hidden) boot partition, install CD or DVD, and start a multicast restore process with Bonjour enabled. All that's needed is the Bonjour name of the multicast server Mac.



Mac HelpMate • Tiger

Hello, mostadmin, your boot disk has 43.8 GB left
SMART Status: Not Supported

Server Data Rate: 6 000000 Port: 7800 TTL: 3
Client Data Rate: 6 000000 Interface: en0
☑ mDNS (Bonjour) Image: /Volumes/Ultratrak/images.
☐ Open in Terminal ASR Multicast @: 234.77.2.123

multicast started at by mostadmin on mostsvr

Multicast Server: asr://mostsvr.local

Figure 3. Mac HelpMate Multicast ASR Settings Tab.

One of the reasons I wrote Mac HelpMate was to make the multicast ASR process easier to configure, tweak, and manage. Note that the ASR URL (asr://mostsvr.local) uses Bonjour to locate the IP address of the server hosting the multicast data stream. Each client simply needs to know that URL, and that's all. With a little extra work, it might be possible to add the ability to browse for multicast streams on the ASR client as well. The mechanism that allows OS X applications and services to use Bonjour is called the mDNSResponder, and it's Open-Source software available for download at http://developer.apple.com/darwin/projects/bonjour. Apple's not only busy building Bonjour into nearly all of its signature applications on OS X and network services on OS X Server, it's also providing an easy way for developers to include Bonjour functionality in their software and hardware products. Network-enabled products can easily use the Bonjour source code to locate servers and peers. Somewhat logically, the mDNSResponder uses UDP (User Datagram Protocol) port 5353, while traditional DNS uses TCP port 53. A quick line in the Terminal shows it, ears pricked, listening for connections:

```
Dual2ghz:~ dean$ netstat -a|grep -i mdn
udp4       0      0  *.mdns                  *.*
```

And a quick grep of /etc/services reveals:

```
Dual2ghz:~ dean$ cat /etc/services|grep 5353
mdns             5353/udp    # Multicast DNS
mdns             5353/tcp    # Multicast DNS
```

One of the first large Open-Source projects to use the Apple mDNSResponder source code is KDE (K Desktop Environment, http://www.kde.org/) 3.4 for Linux. KDE is a very popular Desktop, Office and Application bundle for Linux workstations. Although there are plans for integrating Apple's Bonjour source code more tightly into KDE, for now the best benefit is the discovery of shared X Windows desktops (something that OS X has yet to offer beyond simple screen sharing).

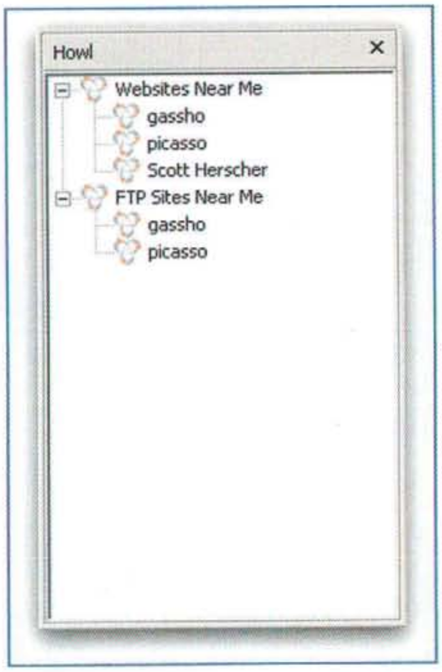


Figure 4. Howl Service Browser for Linux.

Interestingly enough, the KDE developers actually had a *choice* of which mDNSResponder to use. Besides Apple's own Bonjour, there's also the parallel Open-Source Howl project, which implements almost exactly the same functionality for OS X, Windows, and many types of Unix, but with a GPL (GNU Public License) and its own code base. Interestingly enough, Howl (http://www.porchdogsoft.com/products/howl) also offers Zeroconf for OS X, which is available in the form of a Fink installer package (http://fink.sourceforge.net).

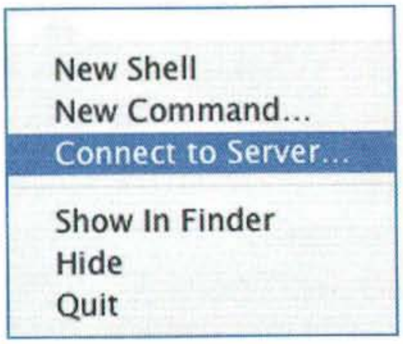


Figure 5. "Connect to Server..." in Terminal Dock Menu.

The Howl browser window is strangely reminiscent of the Network Browser application that shipped with Mac OS 8.5 and 9. However, the Network Browser used SLP (Service Location Protocol), an older standard mostly used by Novell. While it's not completely obvious, such a browser exists in OS X, hidden in the Dock Menu of the Terminal application. To access the browser, hold down the control key and click on the Terminal icon in the Dock or use the right button of a two-button mouse and wait for the "Connect to server" dialog to appear.

MACTECH®

reseller network

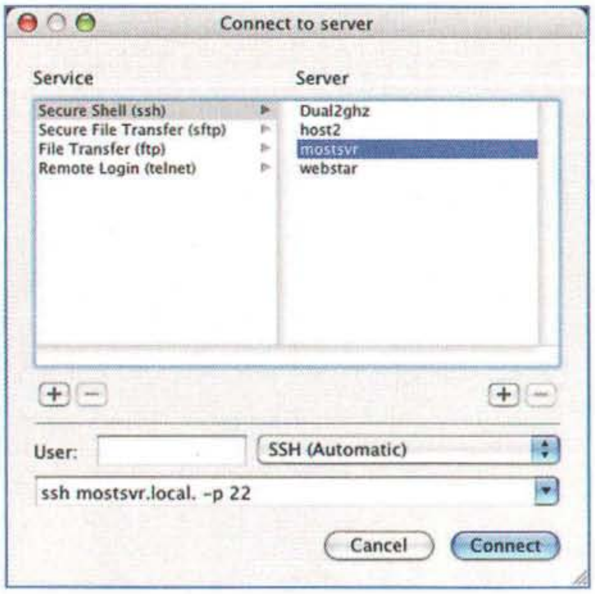
MacTech Helps You Profit!

Let the MacTech Reseller Network put you in the red-hot domain name registration and Internet business.

We've done the heavy lifting — product development, customer support, and infrastructure. All you need to do is sign up!

www.mactechdomains.com/resellers/

So take your business live on the Internet and start earning money TODAY!



```
ssh mostsvr.local. -p 22
```

Figure 6. "Connect to server" Dialog.

It's quite interesting to note that the only appearance of a "browserish" interface is here, for services that many would never truly consider candidates for Bonjour-ization, but for those network admins that might have kept a sheet of manually assigned IP addresses at their desk to ssh into the Macs they manage, it's most welcome! For those on networks with DHCP, a ping scan of the subnet, followed by a reading of the arp cache and a script to pin machine names onto the Ethernet addresses was necessary:

```
Dual2ghz:~ dean$ ping -c 1 192.168.0.255 ; arp -a
PING 192.168.0.255 (192.168.0.255): 56 data bytes
64 bytes from 192.168.0.87: icmp_seq=0 ttl=64 time=0.149 ms

-- 192.168.0.255 ping statistics --
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.149/0.149/0.149 ms
? (192.168.0.1)   at 0:0:94:83:34:68 on en0 [ethernet]
? (192.168.0.84)  at 0:3:93:be:42:ba on en0 [ethernet]
? (192.168.0.86)  at 0:5:2:71:0:d5 on en0 [ethernet]
? (192.168.0.255) at ff:ff:ff:ff:ff:ff on en0 [ethernet]
```

Also interesting is the ability to ping the mDNSResponder on each Mac OS X computer by using the multicast broadcast address:

```
Dual2ghz:~/Desktop dean$ ping 224.0.0.251
PING 224.0.0.251 (224.0.0.251): 56 data bytes
64 bytes from 192.168.0.87:  icmp_seq=0 ttl=64 time=0.735  ms
64 bytes from 192.168.0.128: icmp_seq=0 ttl=64 time=107.39 ms (DUP!)
64 bytes from 192.168.0.86:  icmp_seq=1 ttl=64 time=0.435  ms (DUP!)
64 bytes from 192.168.0.85:  icmp_seq=1 ttl=64 time=0.505  ms (DUP!)
```

Note that the response is from "normal" internal IP addresses, not from the multicast or the Link-Local address. In this respect, mDNSResponder acts as a "helper" in locating the IPv4 address in the routing tables of those computers running Bonjour on the local subnet. And, if that were all Bonjour were capable of, that might be good enough.

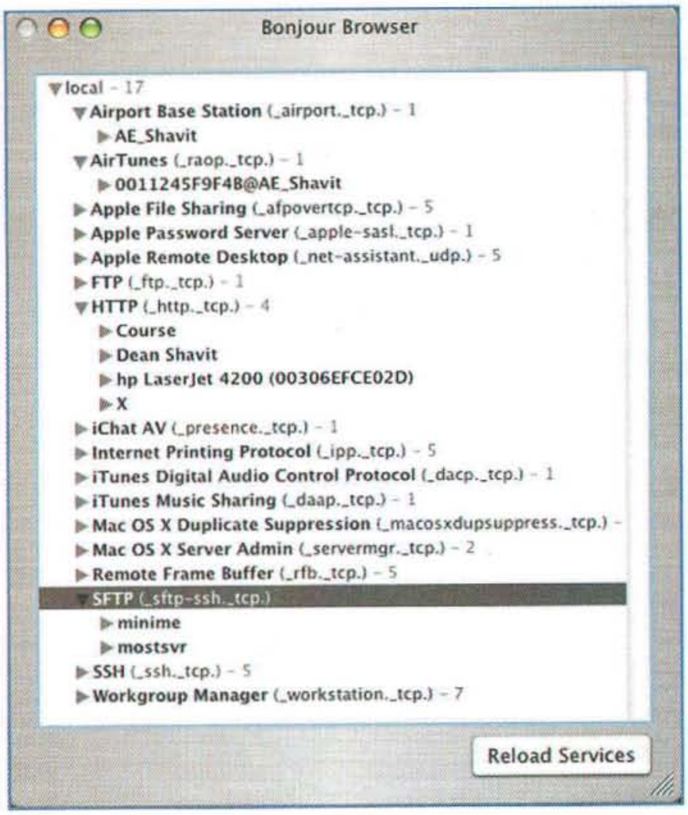
DNS-SD

While the mDNSResponder could be considered to be the engine that connects the multicast, Link-Local, and IPv4 addresses, making it easy for unconfigured computers to talk to each other as well as the configured computers on their local subnet, it's the DNS-SD (Service Discovery) part of Bonjour (and Rendezvous, its predecessor in name) that really brings Bonjour technology into the spotlight (symbol crash) of an OS X user's daily interaction with their network.

When OS X 10.0 was released, SLP (Service Location Protocol) and AppleTalk were used to advertise file sharing resources on the local network. While SLP was a decent method of browsing for servers, there were some significant problems. Configuring the SLP DA (Directory Agent) for the local network wasn't very straightforward, not to mention the infamous issue with laptops broadcasting multiple instances of their presence on the local network for ***each IP address they'd every had for any location they'd ever visited***. The solution was to delete the /var/slp.regfile which contained the SLP registrations for each of those IP addresses.

OS X 10.3 marked a turning point in the fact that SLP was deprecated in favor of mDNSResponder and Rendezvous. Today, Tiger Servers configured as Open Directory Masters, it's even possible to manipulate the Bonjour browsing views of managed clients. This capability, though not very well documented at this point as far as how the backend tech works, is significant in that it uses a managed DNS-SD capability to control what OS X clients see when browsing. Not only can the Managed Network Views be carved up into virtual and logical segments, it's even possible to control ***which*** services the clients can browse for. DNS-SD browses for and announces registered service types.

Some IT managers with a semi-paranoid security focus are of the opinion that the DNS-SD services advertised by Bonjour can constitute a security risk. Although OS X doesn't provide a global browser for all computers on the network, there is a freeware utility called "Bonjour Browser" available at http://www.tildesoft.com/Programs.html that scans the network for registered Bonjour services and displays them in a window. Although it's not of much practical use, it does give away which machines are listening for different types of connections (file services, ssh, etc.) allowing for potential evildoers to get quick down-and-dirty looks at who's offering what network services. Security concerns aside, it's absolutely amazing to see just how deeply integrated Bonjour is in OS X and now diverse the DNS-SD services it supports have become.



```
▼local - 17
  ▼Airport Base Station (_airport._tcp.) - 1
    ▶AE_Shavit
  ▼AirTunes (_raop._tcp.) - 1
    ▶0011245F9F4B@AE_Shavit
  ▶Apple File Sharing (_afpovertcp._tcp.) - 5
  ▶Apple Password Server (_apple-sasl._tcp.) - 1
  ▶Apple Remote Desktop (_net-assistant._udp.) - 5
  ▶FTP (_ftp._tcp.) - 1
  ▼HTTP (_http._tcp.) - 4
    ▶Course
    ▶Dean Shavit
    ▶hp LaserJet 4200 (00306EFCE02D)
    ▶X
  ▶iChat AV (_presence._tcp.) - 1
  ▶Internet Printing Protocol (_ipp._tcp.) - 5
  ▶iTunes Digital Audio Control Protocol (_dacp._tcp.) - 1
  ▶iTunes Music Sharing (_daap._tcp.) - 1
  ▶Mac OS X Duplicate Suppression (_macosxdupsuppress._tcp.) -
  ▶Mac OS X Server Admin (_servermgr._tcp.) - 2
  ▶Remote Frame Buffer (_rfb._tcp.) - 5
  ▼SFTP (_sftp-ssh._tcp.)
    ▶minime
    ▶mostsvr
  ▶SSH (_ssh._tcp.) - 5
  ▶Workgroup Manager (_workstation._tcp.) - 7
```

Figure 7. Bonjour Browser Application Window.

It's pretty safe to say that Apple has a registered DNS-SD protocol for each major service that runs on OS X and OS X Server, as well as at least one to advertise the presence of any Apple hardware product with a network port, such as XServe RAIDs and Airport base stations. DNS-SD registrations are in the following format: _servicename._tcp or _servicename._udp. A complete list of DNS-SD registrations, including instructions for developers on how to register a service for their application or hardware devices, is available at the DNS-SD project website: http://www.dns-sd.org/ServiceTypes.html.

It's always interesting when there's a mistake in the man page of a command in OS X. Sometimes, those errors are trivial, other times they are not so trivial. For example, the man page for the asr command lists the suggested default multicast address as 224.0.0.123. However, as we learned from experience, and this article, Bonjour reserves that network range for its own use, and using that same network range for multicast asr is a very bad idea (think network crash). In a similar, but less critical vein, the man page for the `dns-sd` command line tool suggests that the command first appeared in OS X 10.3, when in reality, it's the `dns_sd.h` header library that's being referenced, as the command line tool is nowhere to be found on an OS X 10.3 box. However, copying the /usr/bin/dns-sd tool from and OS X 10.4 installation will work just fine. The dns-sd command line tool is intended as a helper utility for developers wanting to test Bonjour services, but it can also register a service as well as browse for one.

Remember the Mac HelpMate asr multicast window in Figure 3? Well, it's actually possible to browse for the multicast

JobCapture

Automatic Time and Expense Tracking Software

JobCapture is intuitive, user-friendly client-server software that intelligently and automatically tracks working time and activity. Open a document in any application, and it is automatically being timed by JobCapture. This guarantees that *all work activity is recorded*.

- Perfect for ad agencies, design studios, architects, law firms, PR Firms etc.
- Scalable. Flexible.
- Easy to configure.
- Cost effective. Adding additional users is easy and inexpensive.
- Tracks time and expenses.
- Client and Server software is included.
- Easily customized.
- Real-time data delivery.
- Customizable reporting.
- View data numerically or graphically.
- Import/Export data to SQL, mySQL, XML etc.

JobCapture will *increase productivity and your bottom line* by providing accurate records for billing and analysis. *Stop losing money now— Call today* for more information: 973.763.9494. Reps are available 10:00am-5:30pm, EST.

CaptureWorks

www.captureworks.com

server using the dns-sd command line tool, even if the Bonjour name of the server is unknown. Here's how it works. First, it's necessary to know the registration name of the dns-sd service. For multicast ASR, it's simply asr:

```
Dual2ghz:~/Desktop dean$ ./dns-sd -B _asr._tcp
Browsing for _asr._tcp
Timestamp    A/R Flags if Domain  Service Type   Instance Name
6:49:28.981  Add    2  4 local.  _asr._tcp.     mostsvr
```

Or, for Apple File Servers, it's afpovertcp:

```
mostsvr:~ mostadmin$ dns-sd -B _afpovertcp._tcp
Browsing for _afpovertcp._tcp
Timestamp     A/R Flags if Domain        Service Type           Instance Name
 6:54:10.451  Add     3  4 local.        _afpovertcp._tcp.       mostsvr
 6:54:10.454  Add     3  4 local.        _afpovertcp._tcp.       Dual2ghz
 6:54:10.454  Add     3  4 local.        _afpovertcp._tcp.       host2
 6:54:10.454  Add     2  4 local.        _afpovertcp._tcp.       Panther
```

It's also possible to register an instance of a service on the network with DNS-SD as well:

```
mostsvr:~ mostadmin$ dns-sd -R "My Test" _http._tcp . 80
   path=/path-to-page.htm
Registering Service My Test._http._tcp port 80 path=/path-to-
   page.htm
Got a reply for My Test._http._tcp.local.: Name now
   registered and active
```

I'm Bad, I'm Nationwide

Beyond the gradual adoption of mDNSResponder and DNS-SD technology into other Operating Systems such as Linux, Apple has primed the pump, so to speak, by offering an "official" Bonjour implementation for Windows. I use Apple's Bonjour for Windows whenever setting up Bonjour-capable printers with Windows XP and 2000 computers. It's far simpler than creating a network printer attached to a server or a local TCP/IP port on the Windows box; it's even easy enough for an end user! It's a little known fact, however, that Bonjour for Windows enables the full suite of Bonjour capabilities, not just printers.

Figure 8. Bonjour Logo.

Perhaps the biggest (and most unheralded) change in Bonjour, besides the name from Panther to Tiger was the addition of wide-area (and multiple subnet) DNS-SD advertising via Dynamic DNS Updates and unicast DNS queries (multicasts aren't allowed on the Internet with a few exceptions). Last month, (MT 21.10) I wrote about emerging capabilities for more complex Open Directory deployments via LDAP OUs (Organizational Units) and DACs (Directory Access Controls). As Directory Service deployments get more complex and spread out over wide-area networks, it's also necessary that the discovery protocols for browsing those networks keep developing. As Open Directory matures, Bonjour must keep pace.

Even before Bonjour officially supported multiple subnets or wide-area service discovery, there were attempts to bridge Rendezvous over two network segments. The Rendezvous Proxy project (http://ileech.sourceforge.net/) originally sought a way to allow sharing of iTunes music libraries between subnets, but people found out pretty quickly it could be used to advertise printers, servers, or just about any other Rendezvous service. The developer even released a Rendezvous proxy for Windows. Configuring wide-area support for Bonjour at this time is still somewhat experimental, however, Apple has provided a few white papers detailing the theory behind running a Bonjour DNS Domain over the Internet, as well as specific instructions at www.dns-ds.org.

In Next Month's Source Hound

In next month's column, I'll actually set up a wide-area Bonjour DNS Server, see how well it works, then look to see if it's possible to integrate our new DNS capabilities into Apple's Open Directory LDAP server, instead of using BIND. I'll either be singing like the ZZtopsters, or maybe singing soprano, depending on how it goes.

MT

About The Author

Dean Shavit is an ACSA (Apple Certified System Administrator) who loves to use a Mac, but hates paying for software. So each month he's on the hunt for Open-Source or freeware solutions for OS X. Besides surfing for hours, following the scent of great source code, he's a partner at MOST Training & Consulting in Chicago, where he trains system administrators in OS X and OS X Server, facilitates Mac upgrade projects for customers, and writes for his own website, www.themachelpdesk.com. Recently, he became the surprised father of an application: Mac HelpMate, available at www.machelpmate.com. If you have questions or comments you can contact him: dean@macworkshops.com.

YOUR ULTIMATE 3D AQUARIUM!

NEW
RELEASE!
Allume Systems
AQUAZONE
SEVEN SEAS DELUXE
AQUAZONE SEVEN SEAS DELUXE
Includes Sharks!
40 fish inside! Have as many onscreen as you like!

FORTY BEAUTIFUL FISHES,
TURTLES, JELLIES, & SHARKS!
INTERACTIVE SCREENSAVER
TAP THE GLASS, FEED THE FISH

Allume Systems
A Division of Smith Micro Software

virtual aquarium so vibrant, you'll swear the fish are real! Available from your favorite retailers and catalogs.

Aquazone Seven Seas Deluxe includes 40 fresh and saltwater fish, jellyfish, adult and baby sea turtles, and sharks! Have entire schools of fish on screen swimming through 20 customizable 3D tanks!

Using Open Source Tools to Filter Email on Mac OS X Server

By Paul T. Ammann

If you are a systems administrator, you know that mail servers worldwide are saturated by virus traffic. There are simple, powerful ways to add virus filtering to a Mac OS X Server-based mail server, in order to block, remove, or disable harmful content in incoming and outgoing email. Because Mac OS X Server is a UNIX-based system, there is a wealth of freely-available open-source software that you can download and use, including virus-filtering tools. Keeping your email clean of viruses is always a good idea, especially if you have a heterogeneous network—infected email can be passed through your server to infect machines running other operating systems. With the tools discussed in this article, keeping your email virus-free is easy, free and customizable.

This article steps you through the installation and setup of some widely-used tools: Perl-based Amasvisd-new, Spam Assassin, and ClamAV. There are other, similar open-source tools available, but completing this exercise shows you one way to keep your mail server clean of viruses; you can try other solutions, as well.

Installing Amavisd-new

Amavisd-new is an interface that runs between Postfix, the mail server software that comes with Mac OS X Server, and the virus filtering software that we will download and install. Amavisd-new accepts messages from Postfix and passes them along securely to any of several available virus filters. Here, we will configure Amavisd-new to work with the ClamAV scanner (also available for download at no cost); but the technique discussed in this article can be used with other virus-scanning tools as well.

To get Amavisd-new, which is Perl-based, running on Mac OS X Server, the first step is to download a bunch of Perl modules that it requires. This is most easily done from the command line, via CPAN:

```
sudo zsh
cpan
```

Note: These instructions have you switch to the z shell, you can choose to use another shell if you prefer.

Then, all on one line:

```
install Archive::Tar Archive::Zip Compress::Zlib
Convert::UUlib Digest::MD5 Digest::SHA1 Net::Server
Net::SMTP Time::HiRes Unicode::Map Unicode::String
Unix::Syslog Mail::SpamAssassin
```

This takes a fair while to process. We watch the terminal to monitor progress and to watch for any errors. One package needs extra encouragement:

```
force install Convert::TNEF
```

And finally, when everything is downloaded and installed:

```
quit
```

Note: If CPAN hasn't previously been configured, it will ask for setup details at the beginning of the session. You can either agree to follow and install any prerequisites that may be requested, or you can type `o conf prerequisites_policy` follow at the beginning of the session to do so automatically.

Then we download another prerequisite package, MIME-Tools 6.2, which as of this writing isn't available through CPAN. We untar, build, and install it:

```
sudo zsh
tar xzvf MIME-tools-6.200_02.tar.gz
cd MIME-tools-6.200_02
perl Makefile.PL
make
make install
```

Next, we need to install the Berkeley DB. Mac OS X Panther ships with the Berkeley DB version 1.8, but for this process, we need a newer version, 2.0 or later. If you don't install version 2.x or newer, you will get an error message later on when you are installing the Perl modules.

Download the newest version 4.2.52 source (or whichever version you choose, 2.x or later), from Sleepycat Software.

Copy the tar file to your working directory and untar:

```
# cp db-4.2.52.NC.tar.gz /usr/local/src
# cd /usr/local/src
# tar -zxvf db-4.2.52.NC.tar.gz
```

Go to the build directory and perform the standard build ritual:

```
# cd db-4.2.52.NC/build_unix
# ../dist/configure —prefix=/usr
# make
# make install
```

This should install BerkeleyDB, including the libraries needed for the BerkeleyDB.pm Perl module.

All the prerequisites are now in place. We now use Workgroup Manager to create a user and group, with the former in the latter, both named `amavisd`. Amavisd-new will run as this user, because running it as root is unsafe and best avoided. We assign the user a home directory of `/private/var/amavisd`, and adjust that directory's permissions from the command line:

```
chown amavisd:amavisd /private/var/amavis
chmod 750 /private/var/amavisd
```

Amavisd-new also wants a directory in which it can quarantine infected email:

```
mkdir /private/var/virusmails
chown amavisd:amavisd /private/var/virusmails
chmod 750 /private/var/virusmails
mkdir ~amavisd/tmp ~amavisd/db
```

Now we are ready to install Amavisd-new. Still as the superuser, download the latest stable version and simply untar it in my working directory. Then place the Perl executable in /usr/local/bin:

```
cp amavisd /usr/local/bin
```

and place its config file in the /etc directory:

```
cp amavisd.conf /etc
```

Next we need to edit `/etc/amavisd.conf` just a bit to reflect the setup of our server. We find the line defining the `$myhostname` variable, and set it to our server's hostname. Likewise, we set `$daemon_home` and `$daemon_group` both to 'amavisd', the name of the user and group under which the daemon Amavisd-new will run.

There are numerous other settings in this file, which it's well worth familiarizing oneself with before rolling out a production installation of Amavisd-new, but the default settings work for now.

We can briefly test that the installation has worked, by becoming the amavisd user and running it in debug mode, which displays verbose status messages:

```
su amavisd
/usr/local/bin/amavisd debug
```

It gives us a scroll of output as it loads all its Perl modules, looks

MacResource
the.mac.resource

Wireless Products

Airport Cards	
Extreme:	$69.99
Standard:	$139.99
Routers:	$59.99

Accessories

Apple Pro Speakers: $34.99
Apple Mice or Kbds $19 to $59
Kensington 4-btn Mice: $19

1-888-Mac-Resource

Parts for Every Mac System!

Logic Boards

G3 B&W Rev2: $99
G4 PCI : $129
G4 AGP Graphics: $219
G4 Gigabit: $349
G4 Digital Audio: $329
G4 Quicksilver: $499
G4 Mirrored Doors: $489
iMac Logics from $199
Beige G3 Series: $49

Power Supplies

Beige G3 Desktop: $49
Beige G3 Tower: $69
Blue & White Tower: $89
G4 Quicksilver: $179
G4 Digital Audio: $189
G4 Gigabit Ethernet: $199
G4 AGP Series 237w: $219
G4 AGP Series 208w: $179
G4 Mirrored Doors: $199

Logic board and power supplies require exchange.

Thousands of parts in stock for all Mac systems!

www.mac-resource.com

Mac Systems

G4/733Mhz: $699
G4/1000Mhz: $899
G4/1.25 OS9: Call

New Systems Arriving Daily! Call for latest Stock.

eMac 800Mhz: $529
eMac 1GHz: $659

iMac G4/800 15": $599
G4/1.0Ghz 17": $899

G4 Cubes In Stock!

G3/400 Only $99!
G4/450 Only $299!

iBook G3 700/900 Combo 12": $499/$649
iBook Stock Arriving Daily!

Displays

Apple 17-inch Studio LCD: $349
Apple 23-inch HD Cinema: $1399
Apple 22-inch Cinema: $999
Apple 17-inch Studio CRT (VGA): $99
Apple 17-inch Studio CRT (ADC): $99

Products are open box, refurbished or demo call for more information for scanners, and finally says "parent ready for children." At this point we stop the running process with Control-C, and proceed.

Amavisd-new is installed, but it is just a framework for virus scanning: there is as yet no actual scanner in place to do the job. Amavisd-new can interface with any number of scanners—a complete list is given at the end of the `amavisd.conf` file. If a site already has a license to use a particular compatible scanner, Amavisd-new can handle it. Here, we use the free scanner ClamAV.

Installing ClamAV

The scanner ClamAV is a free antivirus package that includes several useful features, including easy integration with mail servers, and scriptable automatic updating of its virus database. Let's describe how to install and use it to detect and filter mail viruses.

Here is how to install ClamAV. It will run constantly to process the data that Amavisd-new feeds to it. Running ClamAV as root is dangerous and leaves my server open to the risk of intrusion, so instead, we run it as the user amavisd, which we created in the previous section.

In order to verify the digital signatures of updates to the virus library (and hence prevent forgeries and errors), ClamAV needs GMP, the Gnu Multiple Precision Arithmetic Library. We download, unzip, and build this package:

```
sudo zsh
tar xzvf gmp-4.1.2.tar.gz
cd gmp-4.1.2
./configure ; make ; make install
```

(If you have Fink installed on your system, you can install ClamAV even more easily with the Fink command `fink install gmp`).

Next, to install ClamAV, we download the latest stable source package from SourceForge. Untarring it in our working directory as the superuser, we then cd to the source directory, build, and install it, using the familiar process:

```
tar xzvf clamav-0.75.1.tar.gz
cd clamav-0.65
./configure —sysconfdir=/etc —with-user=amavisd —with-
group=amavisd
make
make install
```

The special flags to ./configure set the location of the config file, and the user and group under which ClamAV will execute. After the build process is over, we test the newly created ClamAV executable on a handful of sample viruses that are handily supplied in the source directory:

```
/usr/local/bin/clamscan ./test/test*
```

It finds five viruses:

```
./test//test1: ClamAV-Test-Signature FOUND
./test//test1.bz2: ClamAV-Test-Signature FOUND
./test//test2.badext: ClamAV-Test-Signature FOUND
./test//test2.zip: ClamAV-Test-Signature FOUND
./test//test3.rar: ClamAV-Test-Signature FOUND

———- SCAN SUMMARY ———-
Known viruses: 10131
Scanned directories: 1
Scanned files: 8
Infected files: 5
Data scanned: 0.00 MB
I/O buffer size: 131072 bytes
Time: 1.796 sec (0 m 1 s)
```

It doesn't succeed in finding the virus hidden in a RAR file—to do that, we need to install an unrar too—but since the daemonized version of ClamAV (which is what we are going to be using) can't handle RAR files regardless, that doesn't matter for our purposes.

With that done, we next test ClamAV's automatic database update tool, Freshclam, by typing:

```
/usr/local/bin/freshclam —verbose
```

Freshclam connects to the virus database and downloads any updates that may be waiting:

```
ClamAV update process started at Mon Sep 20 17:25:16
2004
Connected to database.clamav.net (128.121.60.235).
Reading CVD header (main.cvd): OK
Downloading main.cvd [*]
main.cvd updated (version: 12, sigs: 11867, f-level: 1,
builder: tkojm)
Connected to database.clamav.net (128.121.60.235).
Reading CVD header (daily.cvd): OK
Downloading daily.cvd [*]
daily.cvd updated (version: 67, sigs: 97, f-level: 1,
builder: tkojm)
Database updated (11964 signatures) from
database.clamav.net (128.121.60.235).
```

We want Freshclam to run periodically on our system, so we are assured of always having the latest virus definitions. So we add the following line to `/etc/watchdog.conf`, to start Freshclam automatically. The "-c 4" tells it to run four times each day, and the `-u clamav` causes it to run as the clamav user rather than as root:

```
freshclam:respawn:/usr/local/bin/freshclam -c 4 -u clamav
# Freshclam daemon
```

Optionally, we can add the "-l logfile" option to Freshclam, which causes it to log all of its activity to the file specified in logfile.

Putting It Together

Now both ClamAV and Amavisd-new are installed, and ready to work together. If we once again run Amavisd-new in debug mode:

```
su amavisd
/usr/local/bin/amavisd debug
```

it now tells us "Found secondary av scanner Clam Antivirus - clamscan at /usr/local/bin/clamscan"

The team is ready to go. We will keep Amavisd-new running while we test the Postfix setup. We now need to hook our scanning setup into Postfix, so it can handle the mail for our server. We will make two manual changes to the Postfix configuration files. Note that subsequent changes to the mail

JBL®

A Harman International® Company

iPod® OUTSPOKEN.

JBL ON STAGE™: SHARE MUSIC OUT LOUD WITH THIS HIGH-PERFORMANCE SOUND SYSTEM FOR YOUR MP3 PLAYER.

WWW.JBL.COM/JBLONSTAGE

setup made using Server Admin may overwrite these manual changes. First, we stop the Postfix server if it is running, using Server Admin.

Next, we open a new terminal window (leaving Amavisd-new running in the previous one) to edit /etc/postfix/master.cf, and add the following lines to the end of the file:

```
smtp-amavis unix - - n - 2  lmtp
     -o smtp_data_done_timeout=1200

127.0.0.1:10025 inet n - n - -  smtpd
     -o content_filter=
     -o local_recipient_maps=
     -o relay_recipient_maps=
     -o smtpd_restriction_classes=
     -o smtpd_client_restrictions=
     -o smtpd_helo_restrictions=
     -o smtpd_sender_restrictions=
     -o
smtpd_recipient_restrictions=permit_mynetworks,reject
     -o mynetworks=127.0.0.0/8
     -o strict_rfc821_envelopes=yes
     -o smtpd_error_sleep_time=0
     -o smtpd_soft_error_limit=1001
     -o smtpd_hard_error_limit=1000
```

That creates the definition and settings for the Amavisd-new content filter in Postfix. We tell Postfix to re-read its settings:

```
postfix reload
```

and test that Postfix is listening on port 10025:

```
telnet localhost 10025
```

It should say something like:

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 server.local ESMTP Postfix
```

We also check that Amavisd-new is listening on port 10024:

```
telnet localhost 10024
```

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 [127.0.0.1] ESMTP amavisd-new service ready
```

Now we want to tell Postfix to route mail to this newly defined filter. We add a line to the end of /etc/postfix/main.cf:

```
content_filter=smtp-amavis:[127.0.0.1]:10024
max=use=10
```

This instructs Postfix to send all mail to the Amavisd-new filter. Once again, we type:

```
postfix reload
```

to activate the new setting. Now Amavisd-new is filtering all our mail. We send a sample message to my user account on the server, and indeed when we receive the message it contains the header

```
X-Virus-Scanned: by amavisd-new at myserver.com
```

showing that the message has been successfully processed.

Configuration

The README.postfix file that accompanies the Amavisd-new distribution provides detailed information on fine-tuning the setup of Amavisd-new and Postfix. Settings for Amavisd-new can be adjusted in `/etc/amavisd.conf`—in particular, the filter's behavior when it finds a virus can be configured. The default setting places virus-infected email messages in the quarantine directory, `/var/virusmails`. Notification options can be changed, viral mail can be simply deleted or bounced, or all viruses can be redirected to a specified mail account.

ClamAV's settings can be tuned in `./etc/clamav.conf`.

In order to start Amavisd-new running automatically when the server is started, we add the following line to /etc/watchdog.conf:

```
amavisd:respawn:su amavisd -c /usr/local/bin/amavisd
```

```
Then we restart watchdog with:
```

```
sudo killall -HUP watchdog
```

ClamAV doesn't have to be started separately, as it is triggered by Amavisd-new when needed.

The log files for any captured viruses are found in `/var/log/syscan.log`.

Resources

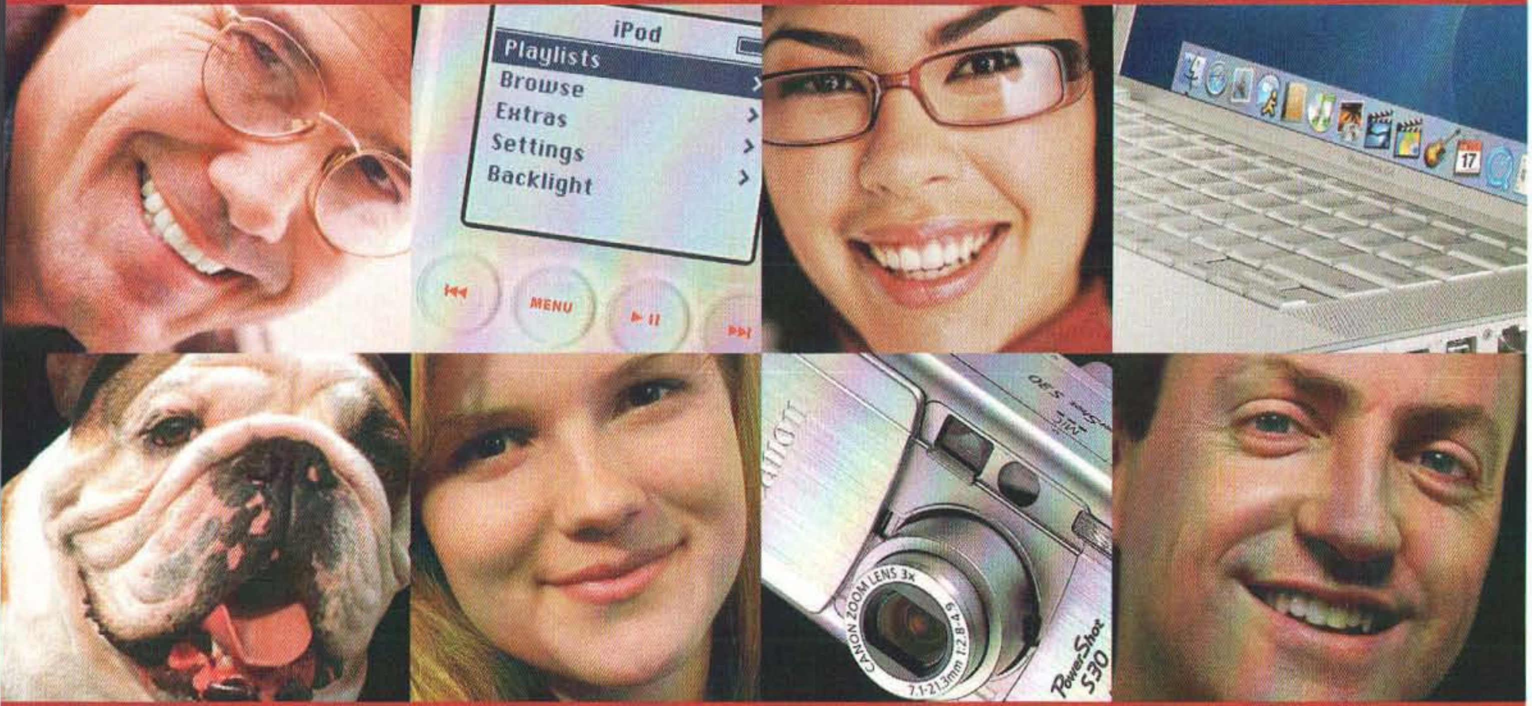
MIME-tools-5.417
http://search.cpan.org/dist/MIME-tools/
Sleepycat Software
http://www.sleepycat.com/download/index.shtml
Amavisd-new, the latest stable version
http://www.ijs.si/software/amavisd/#download
ClamAV
http://www.clamav.net/
ClamAV, the latest-stable version
http://prdownloads.sourceforge.net/clamav/
Gnu Multiple Precision Arithmetic Library (GMP)
http://www.swox.com/gmp/index.orig.html
Fink
http://fink.sourceforge.net/
Unrar
http://unrarx.sourceforge.net/

MT

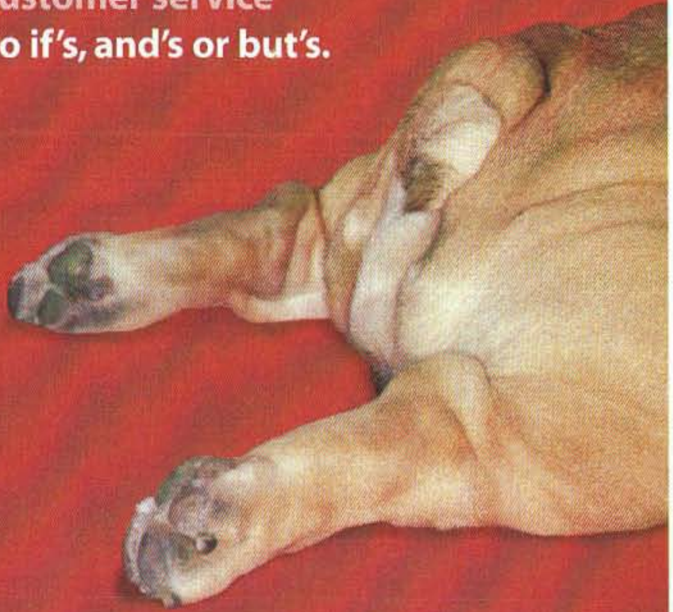
About The Author

Paul T. Ammann has been working in IT for almost 20 years now. He is happily married to his wife Eve for 6 years. He finds writing the author's bio the toughest part of the article.

At Small Dog Electronics, happy customers are our highest priority.



When you shop at Small Dog Electronics, you get more than just great selection and low prices; you also get personalized service from genuine Apple Professionals who take customer service very seriously. And that's a promise... **no if's, and's or but's.**



Small Dog Electronics
www.smalldog.com

Apple Specialist

1-800-511-MACS

A socially responsible business since 1996

STATE PROPERTY

WORKING WITH QUICKTIME PROPERTIES AND PROPERTY LISTENERS

In the previous QuickTime Toolkit article ("The Informer" in MacTech, October 2005), we took a look at the QuickTime metadata functions, which are a replacement for the existing user data functions. We saw that we can use the functions QTMetaDataGetItemProperty and QTMetaDataSetItemProperty to get and set metadata item properties, and that we can use the function QTMetaDataGetItemPropertyInfo to get information about a metadata item property (such as the type or the size of its data).

Introduction

You should get used to this pattern of function naming: `SetProperty`, `GetProperty`, and `GetPropertyInfo`. That's because, as of QuickTime version 6.4, this is now the preferred pattern of naming for functions that inspect and set properties. QuickTime 6.4 introduced the trio of functions `QTSetComponentProperty`, `QTGetComponentProperty`, and `QTGetComponentPropertyInfo` for working with component properties. It also introduced the functions `QTSetMovieProperty`, `QTGetMovieProperty`, and `QTGetMoviePropertyInfo` for working with movie properties. And QuickTime 7 has accelerated this trend. A quick search through the latest QuickTime header files reveals about a dozen additional triples of property-related functions, from `ICMImageDescriptionGetProperty` (and its siblings) to `MovieAudioExtractionGetProperty` (and its siblings).

We've already seen some of the advantages of this new scheme for querying and setting properties. In the case of movie metadata, we could iterate through all metadata items associated with a movie, for instance, and retrieve and display the values of those items without knowing in advance either the size or the type of those values. That's because we could use the `QTMetaDataGetItemPropertyInfo` function to query the metadata item for that information. Also, QuickTime can add new properties to the target objects (movies, tracks, metadata items, image descriptions, and so on) without having to add new functions to get or set those new properties. And, of course, the advantages of conforming ***all*** these various property accessor functions to a single calling pattern should be obvious. Once we've learned the parameter list for one of the `GetProperty` functions, we've effectively learned it for the other dozen similar functions.

But, for several of these target objects, there is an additional payoff. Not only does QuickTime provide a unified mechanism for getting and setting the object's properties, it also provides a mechanism for our applications to be notified when one of these properties changes. That is to say, we can install a ***movie property listener***, an application-defined callback procedure that is executed when the value of a specified property changes. In my mind, this is very important, as it relieves us of the necessity to continually poll the movie for the value of a property we are interested in.

In this article, we'll investigate the movie property functions `QTSetMovieProperty`, `QTGetMovieProperty`, and `QTGetMoviePropertyInfo`, as well as the function to install a movie property listener, `QTAddMoviePropertyListener`.

QuickTime Properties

Let's begin by looking at the declarations of the three principal movie property functions, from the header file Movies.h. QTGetMoviePropertyInfo is declared like this:

```
OSErr QTGetMoviePropertyInfo (
  Movie                     inMovie,
  QTPropertyClass           inPropClass,
  QTPropertyID              inPropID,
  QTPropertyValueType *     outPropType,
  ByteCount *               outPropValueSize,
  UInt32 *                  outPropertyFlags);
```

And the functions QTGetMovieProperty and QTSetMovieProperty are declared like this:

```
OSErr QTGetMovieProperty (
  Movie                     inMovie,
  QTPropertyClass           inPropClass,
  QTPropertyID              inPropID,
  ByteCount                 inPropValueSize,
  QTPropertyValuePtr        outPropValueAddress,
  ByteCount *               outPropValueSizeUsed);
OSErr QTSetMovieProperty (
  Movie                     inMovie,
  QTPropertyClass           inPropClass,
  QTPropertyID              inPropID,
  ByteCount                 inPropValueSize,
  ConstQTPropertyValuePtr   inPropValueAddress);
```

As you can see (and as you may recall from the previous article on movie metadata), we specify a particular movie property by providing a ***property class*** and a ***property ID***. These values, as well as the property value type returned by QTGetMoviePropertyInfo, are all declared as OSTypes:

```
typedef OSType              QTPropertyClass;
typedef OSType              QTPropertyID;
typedef OSType              QTPropertyValueType;
```

In addition, the parameters in which data values are passed or returned are declared in the obvious manner:

```
typedef void *              QTPropertyValuePtr;
typedef const void *        ConstQTPropertyValuePtr;
```

So all we really need to know, to be able to use these functions, are the available property classes and their associated property IDs. The file Movies.h contains definitions of quite a number of constants beginning with "kQTPropertyClass_"; currently, however, most of these classes of movie properties cannot be used with the movie property functions. In this article, we'll look at only two of those classes, for accessing audio and video properties:

```
enum {
  kQTPropertyClass_Audio              = 'audi',
  kQTPropertyClass_Visual             = 'visu'
};
```

Most of the other currently defined property classes (for instance, kQTPropertyClass_DataLocation) and their associated properties are intended for use with the new function NewMovieFromProperties, which we'll investigate in the next ***QuickTime Toolkit*** article. (If perchance we did try to read or write one of those other properties using QTGetMovieProperty or QTSetMovieProperty, we would receive the error code kQTPropertyNotSupportedErr.)

For the class kQTPropertyClass_Visual, these property IDs are defined:

```
enum {
  kQTVisualPropertyID_Hue             = 'vhue',
  kQTVisualPropertyID_Saturation      = 'vsat',
  kQTVisualPropertyID_Brightness      = 'vbrt',
  kQTVisualPropertyID_Contrast        = 'vcon'
 };
```

All these properties can be gotten or set, and all return or take parameters of type Float32.

For the class kQTPropertyClass_Audio, Movies.h contains enumerated constants for just over a dozen properties. However, most of those properties cannot be accessed using QTGetMovieProperty or QTSetMovieProperty. The following properties are accessible using those functions:

```
enum {
  kQTAudioPropertyID_Gain             = 'gain',
  kQTAudioPropertyID_Mute             = 'mute',
  kQTAudioPropertyID_Balance          = 'bala'
 };
```

The gain and balance properties operate on values of type Float32, and the mute property operates on values of type Boolean.

Getting and Setting Property Values

It's quite straightforward to get and set these properties on a QuickTime movie. Consider for example the kQTVisualPropertyID_Brightness property, which governs the brightness adjustment of a movie. This property can takes values ranging from –1.0 (which means the image has no brightness and is hence totally black) to 1.0 (which means the image has maximum brightness and is hence totally white); normal brightness is 0.0. We can programmatically fade a movie to black using the function defined in Listing 1.

Listing 1: Setting a movie's brightness

```
void FadeMovieToBlack (MovieController mc)
{
  Float32      origValue, value;
  Movie        movie = MCGetMovie(mc);
```

// get the current value of the brightness adjustment

```
  QTGetMovieProperty(movie, kQTPropertyClass_Visual,
        kQTVisualPropertyID_Brightness, sizeof(origValue),
        & origValue, NULL);
```

// gradually decrease the brightness to total darkness

```
  for (value = origValue; value >= -1.0; value -= 0.01) {
    QTSetMovieProperty(movie, kQTPropertyClass_Visual,
      kQTVisualPropertyID_Brightness, sizeof(value),
      &value);
   usleep(10000);
   MCDraw(mc, nil);
  }
}
```

Figure 1 shows a QuickTime movie in a movie window, and Figure 2 shows the half-way point of the fading that movie to black.

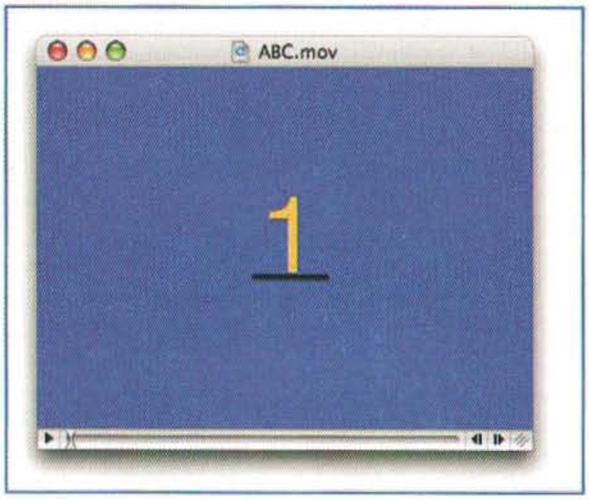


Figure 1: A movie with the default brightness

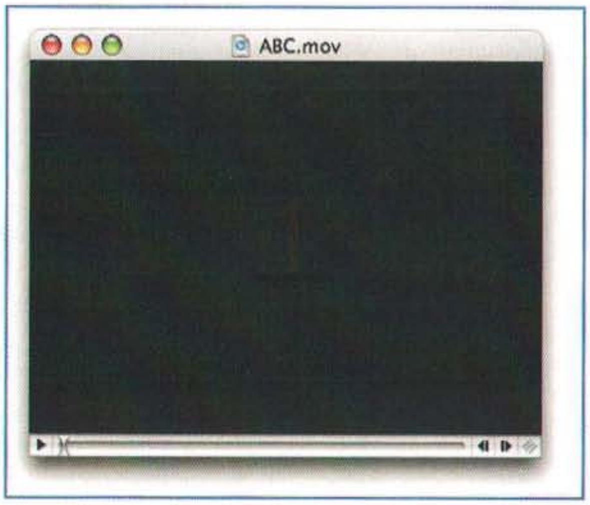


Figure 2: A movie with halved brightness

This is very cool. Even cooler perhaps is to gradually increase the movie's brightness to total white, which we can do by stepping up from the current brightness to 1.0. (Try it; you'll like it.) Keep in mind that the brightness adjustment is only a temporary adjustment and is not saved into the movie file, even if you update the movie file on disk. Ditto for all the other audio and visual properties listed above.

You should also know that the new properties-based APIs are not meant to exclude more classic style APIs. That is, instead of QTGetMovieProperty and QTSetMovieProperty with the specified property class and ID, we could use GetMovieVisualBrightness and SetMovieVisualBrightness. So we could replace the call to QTSetMovieProperty in Listing 1 by this line:

```
SetMovieVisualBrightness(movie, value, 0);
```

The third parameter here is a set of flags, which is currently unused.

Deallocating Property Values

Consider now this question: once we're finished working with a piece of property data, what do we do with it? Surely, the answer depends on the type of value returned by QTGetMovieProperty, which we can determine by inspecting the outPropType parameter in QTGetMoviePropertyInfo. Simple values like Booleans or integers or OSTypes are just copied into the local storage pointed to by the inPropValueAddress parameter and will be cleaned up when the stack frame is destroyed. But what about things like strings or handles or structures?

The answer lies in the QTGetMoviePropertyInfo function. Its last parameter is a pointer to a set of ***property flags*** that contain useful information about the property values returned by a call to QTGetMovieProperty. Currently these flags are defined (in the header file ImageCompression.h):

```
enum {
  kComponentPropertyFlagCanSetLater          = (1L << 0),
  kComponentPropertyFlagCanSetNow            = (1L << 1),
  kComponentPropertyFlagCanGetLater          = (1L << 2),
  kComponentPropertyFlagCanGetNow            = (1L << 3),
  kComponentPropertyFlagHasExtendedInfo      = (1L << 4),
  kComponentPropertyFlagValueMustBeReleased
                                             = (1L << 5),
  kComponentPropertyFlagValueIsCFTypeRef     = (1L << 6),
  kComponentPropertyFlagGetBufferMustBeInitialized
                                             = (1L << 7),
  kComponentPropertyFlagWillNotifyListeners
                                             = (1L << 8)
};
```

As you might guess, the interesting flags right now are the ValueMustBeReleased and ValueIsCFTypeRef flags. If the ValueMustBeReleased flag is set in the returned set of property flags, then we need to release the property value once we are done with it. And if the ValueIsCFTypeRef flag is set, then the returned property value is a reference-counted Core Foundation type and we should release that value by calling CFRelease.

There is one complication here, which is that the values returned in the outPropType parameter to QTGet MoviePropertyInfo are to my knowledge not currently documented. So in cases where the ValueMustBeReleased flag is set but the Value IsCFTypeRef flag is clear, we wouldn't know what routine to call to release the property value. Happily, the values for the property classes and IDs described above are either Float32 or Boolean, which require no deallocation. So for the moment we really don't need to worry about checking those flags.

In the future, however, as the list of gettable and settable movie properties expands, it would be nice to have available a more satisfying way to know how to release the property values. I assume that at that point the relevant data type constants will be exposed. In the meantime, here are a few constants we can use for the principal allocated types:

```
enum {
  kMyTypeCFTypeRef               = 'cfty',
  kMyTypeQDPicture               = 'pict',
  kMyTypeTextHandle              = 'text',
  kMyTypeHandle                  = 'hndl',
  kMyTypeQTAtomContainer         = 'qtat',
  kMyTypeUserData                = 'udat'
};
```

Listing 2 shows a function `ReleasePropertyValue` that we can use to determine whether a property value needs to be released and, if so, to deallocate values of these types. As you can see, we pass in the values we receive from `QTGetMoviePropertyInfo`.

Listing 2: Handling value deallocation

```
void ReleasePropertyValue (QTPropertyClass propClass,
      QTPropertyID propID, QTPropertyValueType propType,
      QTPropertyValuePtr propValueAddress,
      ByteCount propValueSize, UInt32 flags)
{

  // make sure there is storage that must be released

  if (propValueSize == 0)
    goto bail;

  if ((flags & kComponentPropertyFlagValueMustBeReleased)
        == 0)
    goto bail;

  if (flags & kComponentPropertyFlagValueIsCFTypeRef) {

    // we've got a CFTypeRef; release it
    CFRelease((CFTypeRef)propValueAddress);
    goto bail;
  } else {

    // not a CFTypeRef; look for handles and atom containers

    switch (propType) {
      case kMyTypeHandle:
      case kMyTextHandle:
      case kMyTypeQDPicture:
        if (propValueAddress)
          DisposeHandle((Handle)propValueAddress);
        break;

      case kMyTypeQTAtomContainer:
        if (propValueAddress)
          QTDisposeAtomContainer
              ((QTAtomContainer)propValueAddress);
        break;

      default:
        break;
    }
  }

bail:
  return;
}
```

Property Listeners

There is, as far as I can see, no particular advantage to using the new movie property functions instead of specific APIs like `SetMovieVisualBrightness`. The real power of this new set of functions arises from the ability to be informed of changes in movie properties by installing movie property

What's under *your* hood?

High-performance inventions are driven by powerful and robust engines. That's why OpenBase SQL delivers real fault-tolerance—and the horse-power today's multi-user applications need.

Find out how **OpenBase SQL** stacks up against other databases at http://www.openbase.com/databases.pdf

OpenBase Forms™ database GUI building application available soon for OpenBase SQL!

Test-drive OpenBase SQL
with a free, single-user developer license

What will you build with OpenBase?
www.openbase.com/testdrive

"OpenBase allowed us to quit worrying about the database so we could focus on our business."

Josh Paul, Overhyped Technologies,
creator of software that stores film clips for Reality TV shows.

listeners. That is, these new functions allow us to monitor the value of a specific movie property without having to continually poll its value. For instance, we could install a listener for the property class kQTPropertyClass_Audio and the property ID kQTAudioPropertyID_Gain in order to be informed about changes to the movie's volume.

We install a property listener for a specific property class and ID by calling the QTAddMoviePropertyListener function, declared like this:

```
OSErr QTAddMoviePropertyListener (
  Movie                         inMovie,
  QTPropertyClass               inPropClass,
  QTPropertyID                  inPropID,
  QTMoviePropertyListenerUPP    inListenerProc,
  void *                        inUserData);
```

As you can see, we need to indicate the class and ID to be monitored, as well as a pointer to the function to be called when that property changes. We can also pass in a pointer to some application-specific data (that is, a ***reference constant*** or refcon). So, for instance, to listen for volume changes, we could execute this code:

```
QTAddMoviePropertyListener(movie,
     kQTPropertyClass_Audio, kQTAudioPropertyID_Gain,
     (QTMoviePropertyListenerUPP)&PropChangedCallback,
     NULL);
```

The callback function PropChangedCallback might be defined as in Listing 3.

Listing 3: Handling property changes

```
void PropChangedCallback (Movie movie,
  QTPropertyClass propClass, QTPropertyID propID,
  void *refcon)
{
  switch (propClass) {
    case kQTPropertyClass_Audio:
      switch (propID) {
        case kQTAudioPropertyID_Gain:

// do something interesting, like update a volume slider

          break;
        default:
          break;
      }
      break;

    default:
      break;
  }
}
```

You might recall that QuickTime already provides a similar capability in the form of the ***movie controller action filter procedure***, which is informed of actions associated with a movie controller. In a movie controller action filter procedure, we can watch for actions of type mcActionSetVolume and respond accordingly. But there are several important differences between using a movie controller action filter procedure and using a movie property callback function. First, the property callback is executed only when the associated property actually changes value; by contrast, the action filter procedure receives a mcActionSetVolume whenever an mcActionSetVolume command is issued to the movie controller, even if the specified volume is the same as the current volume. Also, mcActionSetVolume is sent to a filter procedure only by movie controller-related calls or user actions; in particular, it is ***not*** sent when Movie Toolbox calls (such as SetMovieVolume) are made. By contrast, the movie property callback is executed for movie controller ***and*** Movie Toolbox volume changes. In both respects, the movie property listening behaviors seems preferable to the movie controller action filter procedure behaviors.

When we are finished listening to a specific movie property, we should unhook the property listener by calling QTRemoveMoviePropertyListener, like this:

```
QTRemoveMoviePropertyListener(movie,
     kQTPropertyClass_Audio, kQTAudioPropertyID_Gain,
     (QTMoviePropertyListenerUPP)&PropChangedCallback,
     NULL);
```

This is stop PropChangedCallback from being called when the volume of the specified movie changes.

One final point: not all movie properties are currently listenable. We can determine whether a specific property is listenable by calling QTGetMoviePropertyInfo and inspecting the set of property flags returned. If the kComponentPropertyFlagWillNotifyListeners flag is set, then that property is indeed listenable.

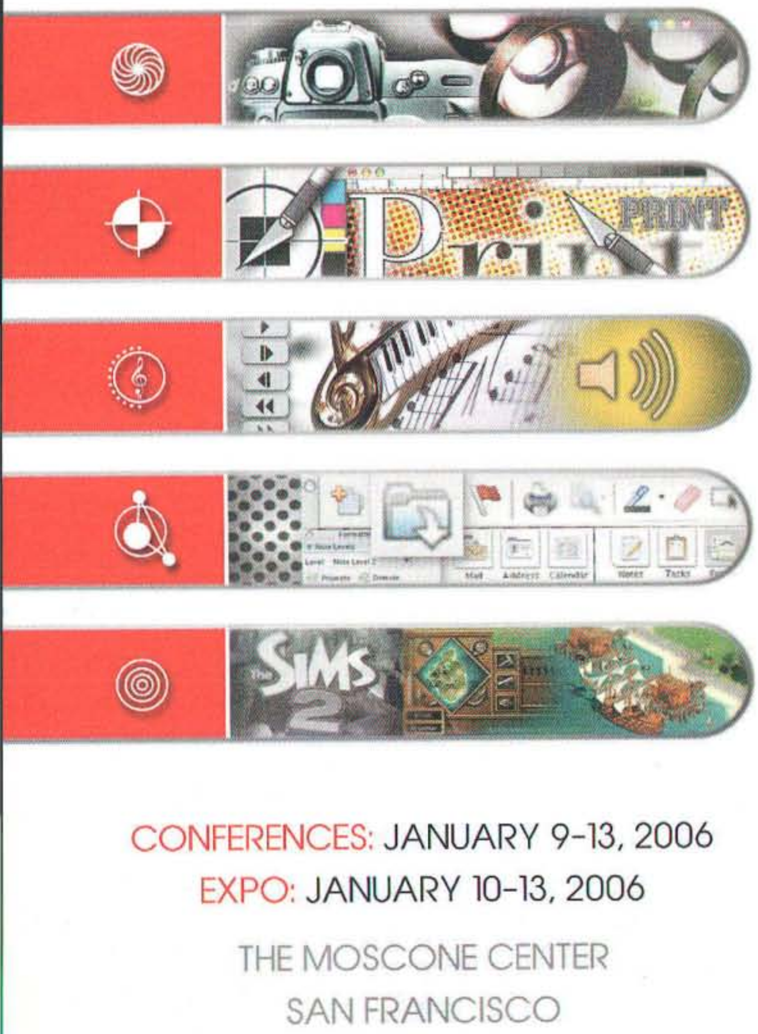
Conclusion

The movie property functions introduced in QuickTime version 6.4 provide a standard way for us to get and set some of the properties associated with QuickTime movies. And, perhaps more important, they provide an easy and reliable way for us to monitor changes in those properties, by installing property listeners. In this article, we've seen how to work with the handful of audio and visual properties that are currently supported by these functions. In the next article, we'll continue working with movie properties, by looking at a new function for opening a QuickTime movie specified by a collection of properties, NewMovieFromProperties.

About The Author

Tim Monroe is a member of the QuickTime engineering team at Apple. You can contact him at monroe@mactech.com. The views expressed here are not necessarily shared by his employer.

Macworld
Conference & Expo®



Macworld Conference & Expo is the #1 event for the Mac Community. Attendees can find the latest, newly announced products and learn to run their Macs more efficiently no matter what their needs are.

Macworld Conference & Expo San Francisco 2006 has your next Mac solution with training to boost your Mac abilities on OS X Tiger, music and audio, video, graphic design, photography, digital imaging, gaming, and more. Recognized industry experts provide professional and comprehensive information, networking opportunities abound and you'll get to meet hundreds of companies supplying the Mac products, innovations and applications that you want to see.

CONFERENCES: JANUARY 9-13, 2006
EXPO: JANUARY 10-13, 2006

THE MOSCONE CENTER
SAN FRANCISCO

M
SF

macworldexpo.com
Register online using Priority Code: D2001 by December 9, 2005 to take advantage of Early Bird Pricing!

Flagship Sponsors:
Macworld Macworld.com

IDG WORLD EXPO

Configuring and Running X11 Applications on Mac OS X

By Paul T. Ammann

The X Window System (more commonly called X11) on Mac OS X provides significant opportunities for Mac OS X developers. Based on the open source XFree86 project, X11 for Mac OS X is compatible, fast, and fully integrated with Mac OS X. It includes the full X11R6.6 technology including an X11 window server, Quartz window manager, libraries, and basic utilities such as xterm. Whether a Unix user or an X11 developer (or both), Mac OS X offers a platform where your applications can run without modification. On a Mac, any of the thousands of available X11 applications can run in a window running concurrently alongside iTunes, Microsoft Excel, and any other Cocoa, Carbon, or Java applications.

There are some things to know about X11 on Mac OS X before you start, and this article outlines the key issues you should be aware of. Many existing X11 applications from the UNIX world are available to use for free—if you know the "secret handshake." That is, you can often easily get the source code, but it's up to you to build and install the product. There are some binary distributions available as well, with applications pre-built for X11 on Mac OS X. This represents a new source of useful software that you don't want to overlook.

Let's first review some X11 basics (for those new to X11), then discuss installing the X11 environment on Mac OS X and starting it for the first time. This section includes a description of the advantages stemming from integration with both the Finder and Quartz. It also discusses differences between Terminal and `xterm`, and full screen support. Next, several X11 configuration issues are covered, including X11 forwarding, configuring `xauth`, using `ssh` to run remote sessions, and PseudoColor support. Then come examples of building X11 applications from source using `configure`, `IMake` and `xmkmf`; and installing binaries using Fink. We conclude with instructions on downloading and running OpenOffice, and point you to further resources for next steps.

X11 Basics

If you're not already familiar with it, X11 can be a bit confusing. You need to understand a few terms and concepts that are essential to X11 before you read the rest of this article. If you are an X11 user, you can skip to the next section.

Which Machine Is the Client?

One important aspect of the X11 architecture is that the typical client and server terminology is reversed. Instead of a user's local client machine asking a remote server machine to do something and send the output back to the client, the user invokes a (potentially remote) client which sends its output back to the user's local X11 display server. To make this work, the user needs to be able to connect to the client, the server must allow display connections from the client, and the $DISPLAY environment variable must be properly set on the client.

How and Where $DISPLAY Variables Are Set

`$DISPLAY` refers to the X11 display server screen. It specifies what display server will receive the output generated by the application run on the client. For a given session, or user, you can specify on which output device the output should appear.

If you use `ssh` to login to the client from an existing xterm, the `$DISPLAY` value will be set automatically, and routed back to the machine from which the connection was initiated.

X11 Installation and Execution

X11 is available as an optional install on the Mac OS X v10.3 Panther, and Mac OS X v10.4 Tiger install disks. Run the Installer, select the X11 option, and follow the instructions. You should install the X11 SDK as well, which is included on the Panther Developer CD. If you intend to download X11 source code and build your own binaries, you will need the tools and headers included in the SDK.

If you have Mac OS X Server, you first need to download the X11User.dmg (Resources). Look for the Download X11 button near the bottom of the content area on this page. This download requires you to login using your Apple ID; if you do not already have one, you can register at this time. After downloading, double-click the package icon to install. You also need to install the X11 SDK in order to build X11 applications.

The Installer puts X11.app into `/Applications/Utilities`. Simply double-click to launch. Congratulations! You now have an X11 environment running on your Macintosh.

Note: Installing X11 on pre-Panther systems requires manually installing XFree86 and XDarwin.app from the Sourceforge "XonX" project (Resources).

X11 and Finder Integration

Apple's X11 implementation is based on the widely-used XFree86 project. The executable that controls the environment, X11.app, runs like any other application in the Finder. You can think of X11.app as a gateway to the X11 environment: if you click the X11.app Dock icon, you enter the X11 world.

X11.app may be displayed as X11 in the Applications/Utilities folder, depending on the Finder preference "Show all file extensions."

A tremendous benefit of this integration is that the Finder responds to a double-click on a UNIX or X11 application icon by starting X11 (if it's not already running) and launching the application. This feature can significantly reduce the time you spend typing command-line launch commands.

X11 and Quartz Integration

The Quartz Window Manager (`quartz-wm`) runs as the default window manager, although you can run alternate window managers if you prefer. A significant advantage of `quartz-wm` is that it integrates Quartz with X11, and adds Aqua buttons (close, minimize, maximize) to X11 windows. This contributes to a unified appearance of visible application windows when running X11 in rootless, or shared screen, mode. In addition, the X11 Dock icon displays running apps in its menu, allowing you to easily bring X11 apps to the front.

`quartz-wm` also provides the pasteboard integration that allows text copying between windows. For example, you can copy text from a Terminal session to an xterm window running under X11. Note that because the key mappings differ between the two environments, you need to use the UNIX equivalents for text manipulation on the X11 side. Command-C copies selected text in both Mac OS X and xterm. But while Command-V pastes in Mac OS X, in X11 the paste key sequence differs across applications. For example, in xterm you use Option-Click to paste the buffer contents into the current xterm window. This simulates clicking with the middle mouse button held down; UNIX systems typically have multi-button mice. Other X11 applications may behave differently. See the X11 FAQ (Technical Q&A QA1232, see Resources) for a more detailed discussion.

xterm vs. Terminal and open-x11

`xterm` presents a much simpler shell window than Terminal. But it provides a significant advantage: when you start an xterm session, it sets up the X11 environment for you. You can then easily run X11 applications from the command-line. By contrast, in Terminal you need to run the `/usr/bin/open-x11` script to set up the X11 environment and launch X11 applications, as shown here:

```
$ /usr/bin/open-x11 /sw/bin/xgalaga
```

On the other hand, `xterm` does not integrate with Mac OS X technologies the way Terminal does. For example, xterm does not support drag and drop: you cannot drop a folder path into xterm to easily change to that directory with the cd command.

Full-screen Support

The default mode is for X11 windows to share the screen with native Mac OS X (Quartz-based) applications. However, there is an option to run X11 in full-screen mode, where all the X11 applications appear on a full-screen X11 root window, and the Mac OS X desktop and toolbar are not visible. This can be useful if you are running a full X11 desktop environment (such as KDE or GNOME), need access to the root window, or simply want to segregate your UNIX and Mac applications.

Important: You can always toggle back to the Mac OS X desktop using Command-Option-A. To re-enter X11, click the X11 Dock icon. If you forget the key sequence, Command-Option-Escape will bring up the Force Quit Applications dialog in the Finder. At this point you are back in Mac OS X, the X11 environment is still running, and you can re-enter at will. You do not need to force quit X11.

X11 Configuration

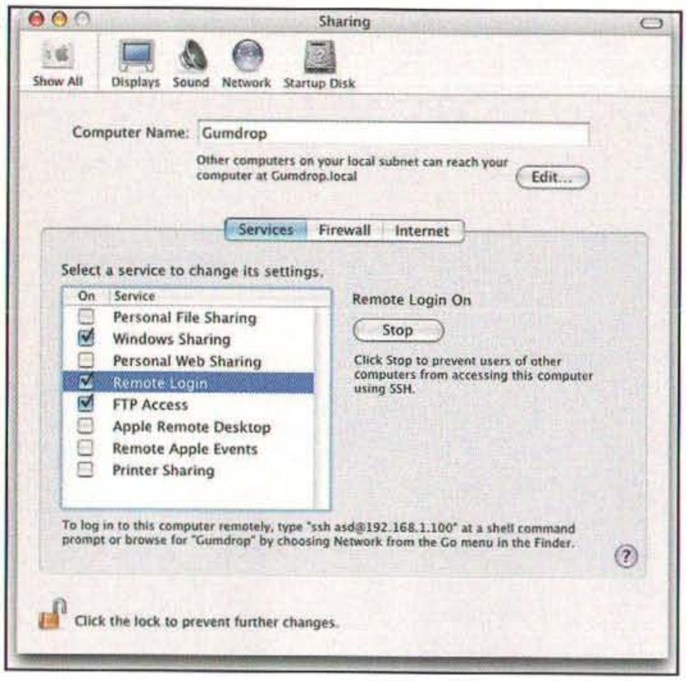
X11 is highly configurable, particularly with regard to security. In addition, older X11 applications that rely on the PseudoColor color mode may need some help to run correctly. Each of these points is addressed in this section.

X11 Forwarding

X11 forwarding allows the X11 connection to be tunneled from the remote system (client) to the local system (display server). For security reasons, Mac OS X does not enable X11 forwarding by default. In order for clients to receive X11 forwarding, the system administrator must explicitly enable it on the Mac OS X system. Enabling X11 Forwarding, Technical Q&A QA1383 (Resources) shows how to perform this task.

Using ssh -X for Remote Session

This section illustrates the use of `ssh -X` to connect from a server to a client. `ssh -X` is preferred over other methods because it encrypts the communication between the server and client. The client is assumed to be running Mac OS X. In order for `ssh -X` to work, you must enable both X11 forwarding as discussed above, and Remote Login on the client (see Figure 1), before attempting to login from the server.



Computer Name: Gumdrop

Other computers on your local subnet can reach your computer at Gumdrop.local

Select a service to change its settings.

On	Service
	Personal File Sharing
✓	Windows Sharing
	Personal Web Sharing
✓	Remote Login
✓	FTP Access
	Apple Remote Desktop
	Remote Apple Events
	Printer Sharing

Remote Login On

Click Stop to prevent users of other computers from accessing this computer using SSH.

To log in to this computer remotely, type "ssh asd@192.168.1.100" at a shell command prompt or browse for "Gumdrop" by choosing Network from the Go menu in the Finder.

Click the lock to prevent further changes.

Figure 1: Enabling Remote Login in the Sharing Preferences

The following sequence walks through the establishment of a connection between the server and client, and running an application. In this scenario, the display server is named Mertz, and the client is named Gumdrop. The X11 user on Mertz wants to connect to Gumdrop, run `xcalc`, and have the calculator display on the primary Mertz screen. The username is `asd` on both systems.

First, login to the client:

```
Mertz:~ asd$ ssh -X -l asd Gumdrop
asd@gumdrop's password:
Last login: Wed Nov 10 13:20:57 2004 from fe80::20d:93ff:
Welcome to Darwin!
[Gumdrop:~] asd%
```

Run `xcalc`:

```
[Gumdrop:/usr/X11R6/bin] asd% ./xcalc &
[1] 2452
[Gumdrop:/usr/X11R6/bin] asd%
```

The calculator will launch at this point and display on machine Mertz. After closing the calculator, logout of the client:

```
[Gumdrop:/usr/X11R6/bin] asd% exit
logout
Connection to Gumdrop closed.
```

`ssh -X` with X11 forwarding is the preferred approach for running remote X11 applications. The next two options, `xhost` and `xauth`, are not as secure. However, we discuss them because they are still used.

Using xhost to Control Server Access

`xhost` controls access to a display server. You run `xhost` on the server to specify which clients may send program output to the server. By itself this does not sound so bad, but `xhost` is not very secure and can leave you exposed. Plus, `xhost` requires more setup than `ssh -X`. The easiest way to use it is the following:

```
xhost +<client-hostname>
```

The hostname is then added to an internal list of clients. That host can now access your display. Because this command is performed on a per-machine basis, every user on the client machine can access the display server. On a network this is an invitation to trouble. Even more dangerous is not specifying a hostname, because then all hosts can access the display.

You can specify a username in place of a hostname. This allows other users on the local machine to access the display server being executed by the current account.

```
xhost <username>
```

The '-' (minus sign) character undoes a setting. For example, to disable access from a particular host:

```
xhost -<hostname>
```

Using the `xcalc` example discussed previously, first add the client to the access list on the server (Mertz).

```
Mertz:~ asd$ xhost +Gumdrop
Gumdrop being added to access control list
Mertz:~ asd$
```

After connecting to the client, set the `$DISPLAY` value on the client to be the primary screen of Mertz:

```
[Gumdrop:~] asd% cd /usr/X11R6/bin
[Gumdrop:/usr/X11R6/bin] asd% setenv DISPLAY Mertz:0.0
Gumdrop:/usr/X11R6/bin] asd%
```

Run `xcalc`:

```
[Gumdrop:/usr/X11R6/bin] asd% ./xcalc &
[1] 2452
[Gumdrop:/usr/X11R6/bin] asd%
```

After closing the calculator and logging off the client, remove the client from the server's list:

```
Mertz:~ asd$ xhost -Gumdrop
Gumdrop being removed from access control list
Mertz:~ asd$
```

Using xauth to Control Server Access

`xauth` adds a greater degree of security than `xhost` by using a cookie generated on the local machine (display server) to control access by the remote machine (client). You generate the cookie, then copy the cookie to the client. When you add the server to the list of hosts known to the client, you pass the cookie as well. When the client connects back to the display server, the cookie is used to authenticate the client.

Learning More about ssh -X, xauth and xhost

A very good discussion of the benefits and pitfalls of `ssh -X`, `xauth`, and `xhost` may be found at the OroborOSX page (Resources).

There are additional options and variations on the `xhost` flags discussed above. More information on `xhost` is available in the man pages (type "man xhost") or on the Internet. Here are a couple of useful links covering `xhost` and `xauth`:

- http://www.leidinger.net/X/xhost.html
- http://www.acm.uiuc.edu/workshops/cool_unix/xauth.html

PseudoColor Support

X11 applications that were written in the days when video memory was relatively scarce may occasionally run into trouble with modern display hardware. X11 supports multiple color models, all of which use pixel values as indices to determine the RGB or grayscale value that will be sent to the video hardware. These models are distinguished by their specification of color vs. grayscale, separate vs. combined RGB colormaps (color lookup tables), and dynamic vs. static entries in each colormap.

PseudoColor is one of the X11 color models. In the PseudoColor model, each frame buffer (video memory) pixel value is used as in index into a single colormap. The entry at that index contains individual red, green, and blue values, which are then sent to the display hardware. This indexing scheme allows applications to access a subset of the available colors for a display. For example, an 8-bit frame buffer that indexes into a 24-bit colormap, containing 8 bits each for red, green, and blue values, supports 256 simultaneous colors, out of a total of 16+ million.

X11 includes additional color models. For example, the DirectColor model uses separate red, green, and blue colormaps. In this case the frame buffer value consists of separate RGB indices into each colormap. Because each colormap is typically 8-bits wide, the number of simultaneous colors or RGB combinations is higher with DirectColor than with PseudoColor.

However, X11 does not support PseudoColor automatically, which presents a problem for applications that require PseudoColor. Here are a couple of solutions:

1. If you want to run an application that requires PseudoColor, and you do not need to simultaneously run other applications that require DirectColor, then you can launch X11 in 256 color mode. To do this, open the X11 preferences, select the Output pane, and change the Colors value to 256. Restart X11, and then launch your application.

2. If you want to run both PseudoColor and DirectColor applications at the same time, you can run a second X server in 256 color mode to handle the PseudoColor apps. This second server must be started from the command-line with the -depth 8 argument, and the -displayID n argument to specify a display other than the default (which is used by X11.app). This allows applications that need PseudoColor support to use the second server and DirectColor apps to use the first server. For example:

Keep Your
Rack Mount
Equipment *Quiet*
And
Locate It
Where You *Need It*

XRackPro2

Noise Reducing
Rack Mount
Enclosure Cabinet

- Apple Approved and Recommended
- Reduces Noise from Multiple Chassises Up to 75%
- Heavy Duty Locking Casters Add Mobility and Stability

GizMac Accessories, LLC.

XRackPro.com

(800) 475-1677

```
Mertz:~ asd$ Xquartz -depth 8 -displayID 1
```

3. There is no support for using both PseudoColor and DirectColor at the same time in a single application.

Building and Installing X11 Software

This section contains examples of downloading and building X11 applications. You have several possibilities, depending on how the code is packaged. The standard UNIX approach for building from source is to first generate a `makefile`, then compile with `gcc`. If the makefile does not exist, you can either use a `configure` script (if provided) or `Imake`. If a binary has already been packaged, you can use a package manager such as Fink to download and install the working application. Each of these options is discussed here.

Generating a makefile with `configure`

First, download and unpack the tarball (the .tar file, or .tar.gz if gzipped). This example assumes you have unpacked the source code for the xpdf application. Since a `configure` script is included, you first run configure from within the project directory to generate the makefile, then `make` to compile, then `make install` to complete the build:

```
Mertz:~ asd$ cd /Downloads/xpdf-3.00
Mertz:/Downloads/xpdf-3.00 asd$ ./configure
checking for gcc... gcc
checking for C compiler default output... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
. . .

Mertz:/Downloads/xpdf-3.00 asd$ make
cd goo: make
g++ -g -O2 -DHAVE_CONFIG_H -I.. -I. -c GHash.cc
g++ -g -O2 -DHAVE_CONFIG_H -I.. -I. -c GList.cc
g++ -g -O2 -DHAVE_CONFIG_H -I.. -I. -c GString.cc
g++ -g -O2 -DHAVE_CONFIG_H -I.. -I. -c gmempp.cc
g++ -g -O2 -DHAVE_CONFIG_H -I.. -I. -c gfile.cc
. . .

Mertz:/Downloads/xpdf-3.00 asd$ make install
mkdir -p /usr/local/bin
/usr/bin/install -c xpdf/pdftops /usr/local/bin/pdftops
/usr/bin/install -c xpdf/pdftotext /usr/local/bin/pdftotext
/usr/bin/install -c xpdf/pdfinfo /usr/local/bin/pdfinfo
/usr/bin/install -c xpdf/pdffonts /usr/local/bin/pdffonts
. . .
Mertz:/Downloads/xpdf-3.00 asd$
```

The output from make install shows that this package installs into /usr/local/bin/.

Generating a makefile with Imake and xmkmf

If a configure script is not provided, you can generate a `makefile` by running the Imake command. But, since Imake requires a number of arguments, so you should instead use the simpler xmkmf command, which packages most of the command-line arguments for you, and then invokes Imake. The following listing uses the game xpacman to illustrate the preceding steps. The ls command provides before and after directory snapshots to show that xmkmf did indeed generate a makefile.

```
Mertz:/downloads/xpacman Folder/xpacman.1 asd$ ls
Imakefile               Makefile.noimake
xpacman.README          xpacman.c

Mertz:/downloads/xpacman Folder/xpacman.1 asd$ xmkmf -a
imake -DUseInstalled -I/usr/X11R6/lib/X11/config
make Makefiles
make: Nothing to be done for `Makefiles'.
make includes
make: Nothing to be done for `includes'.
make depend
gccmakedep  --  -I/usr/X11R6/include    -D__DARWIN__ -
DNO_ALLOCA -DX_LOCALE -DCSRG_BASED
       -- xpacman.c
cc: cannot read specs file for arch `i386'

Mertz:/downloads/xpacman Folder/xpacman.1 asd$ ls
Imakefile                   makefile
xpacman.README
Makefile.noimake           makefile.bak              xpacman.c
Mertz:/downloads/xpacman Folder/xpacman.1 asd$ more makefile
```

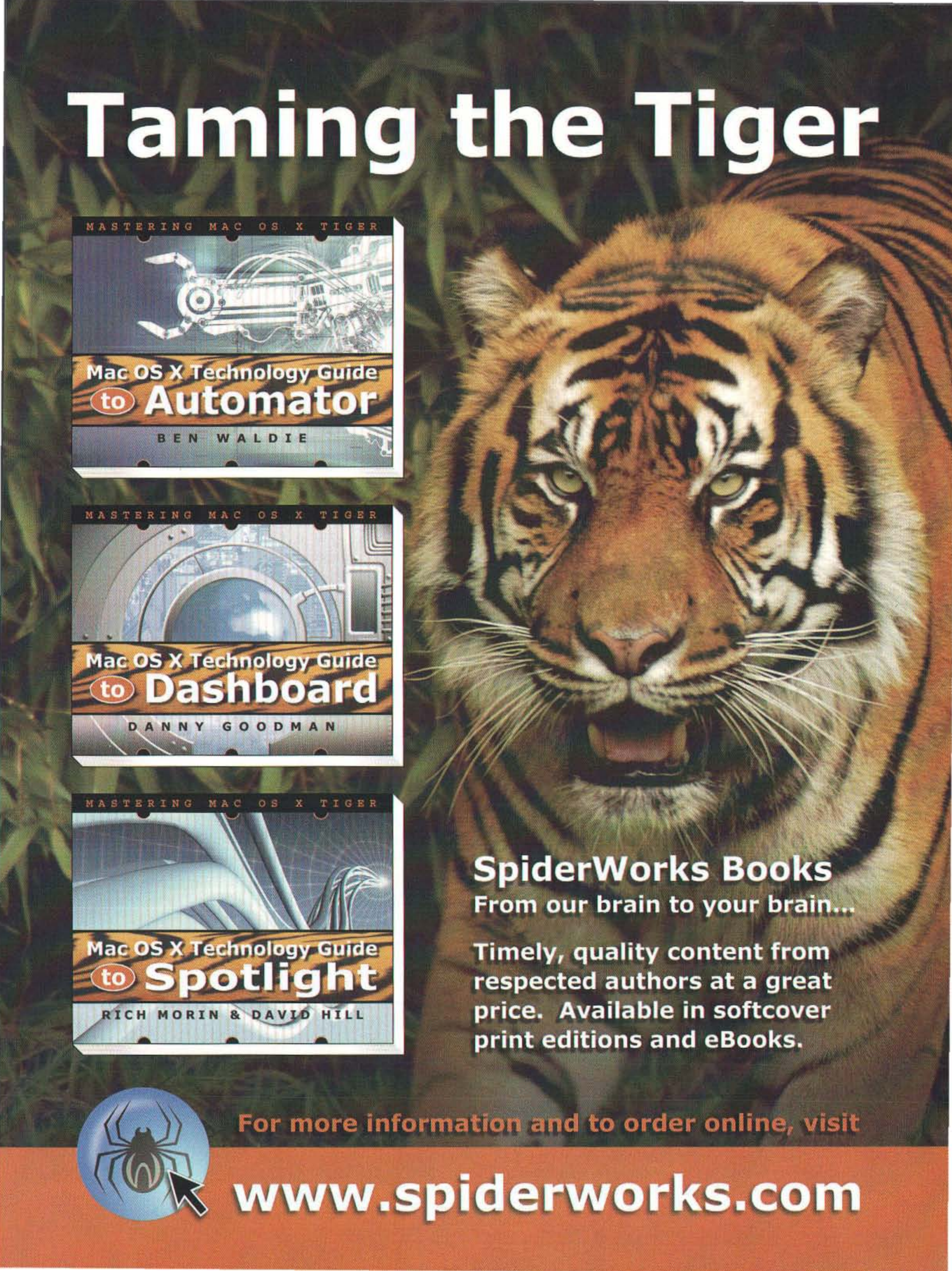
In spite of the cc warning message, a makefile was generated. Now you can run make to compile the program.

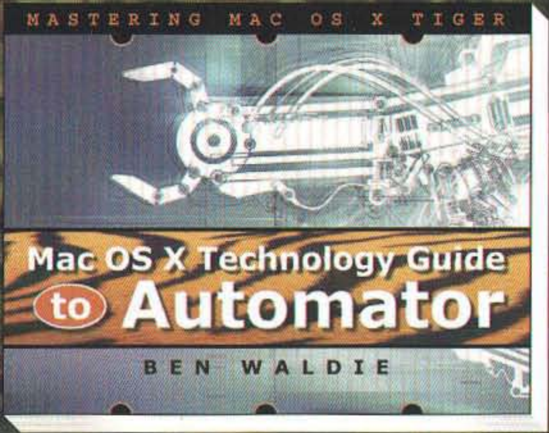
```
Mertz:/downloads/xpacman Folder/xpacman.1 asd$ make
/usr/bin/cc -g -Os -Wall -Wpointer-arith -no-cpp-precomp    -
I/usr/X11R6/include
    -D__DARWIN__ -DNO_ALLOCA -DX_LOCALE -DCSRG_BASED    -
c -o xpacman.o xpacman.c
xpacman.c: In function `main':
xpacman.c:250: warning: suggest explicit braces to avoid
ambiguous `else'
xpacman.c:306: warning: embedded `\0' in format
xpacman.c:449: warning: implicit declaration of function
`tolower'
xpacman.c: In function `setup_maze':
xpacman.c:819: warning: unused variable `cx'
xpacman.c:819: warning: unused variable `cy'
xpacman.c:819: warning: unused variable `w'
xpacman.c:819: warning: unused variable `h'
xpacman.c:819: warning: unused variable `p'
xpacman.c: In function `plot_pacman':
xpacman.c:1074: warning: unused variable `x'
xpacman.c:1074: warning: unused variable `y'
xpacman.c:1114: warning: enumeration value `DEAD' not handled
in switch
xpacman.c:1072: warning: `plotimage' might be used
uninitialized in this function
xpacman.c: In function `update_game_eat':
xpacman.c:1190: warning: implicit declaration of function
`which_ghost_collide'
xpacman.c: In function `newpacpos':
xpacman.c:1207: warning: enumeration value `DEAD' not handled
in switch
xpacman.c: In function `newghostpos_eat':
xpacman.c:1428: warning: enumeration value `DEAD' not handled
in switch
xpacman.c: In function `newghostpos':
xpacman.c:1451: warning: enumeration value `DEAD' not handled
in switch
rm -f xpacman
/usr/bin/cc -o xpacman -g -Os -Wall -Wpointer-arith -no-cpp-
precomp
     -L/usr/X11R6/lib   xpacman.o -lXext -lX11
make: *** No rule to make target `xpacman.man', needed by
`xpacman._man'.  Stop.
Mertz:/downloads/xpacman Folder/xpacman.1 asd$ make install
install -c -o root -g wheel   xpacman /usr/X11R6/bin/xpacman
. . .
Mertz:/downloads/xpacman Folder/xpacman.1 asd$
```

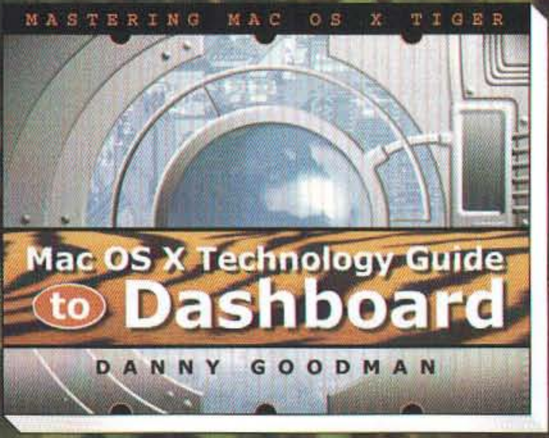
Installing Binaries with Fink and FinkCommander

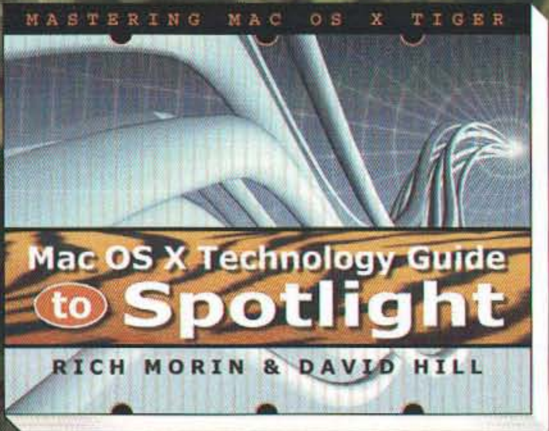
The Fink package manager handles download, installation, and removal of binaries, as long as a package has been provided on one of the Fink servers. In fact, you should try Fink before attempting to build from source (let someone else do the hard work!). You first need to install Fink from SourceForge. Fink differs from typical UNIX installers by installing itself and managed packages into `/sw/bin`. This prevents collisions with

Taming the Tiger









SpiderWorks Books
From our brain to your brain...

Timely, quality content from respected authors at a great price. Available in softcover print editions and eBooks.

For more information and to order online, visit

www.spiderworks.com

other UNIX software, which typically reside in /usr/local/bin.

Fink responds to the `list` command by connecting to its repositories and displaying a list of available packages:

```
Mertz:~ asd$ /sw/bin/fink list
         . . .
         xft2                        [virtual package]
         xft2-dev                       [virtual package]
         xft2-shlibs                    [virtual package]
         xgalaga                    2.0.34-1                  Clone of the
classic game of galaga
         xiangqi                        [virtual package]
         ximian-connector          1.4.7-2                  M$ Exchange
plugin for evolution
         xinvaders                  2.1.2-2                   Space
Invaders clone for X
         . . .
Mertz:~ asd$
```

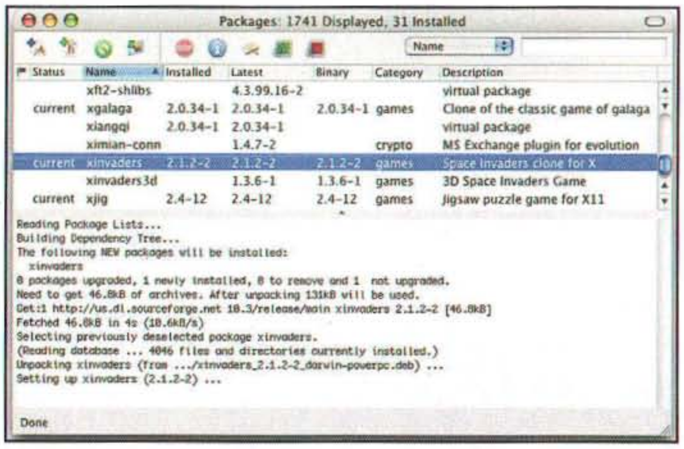
Fink can be used to install or remove packages. When installing, Fink checks for supporting packages that may be needed for the installation. If any are missing, Fink asks if you want to install them, and then handles the download and installation automatically. This example installs the xgalaga package:

```
Mertz:~ asd$ /sw/bin/fink install xgalaga
/usr/bin/sudo /sw/bin/fink  install xgalaga
Password:
Information about 1742 packages read in 1 seconds.
The following package will be installed or updated:
 xgalaga
. . .
Mertz:~ asd$
```

Running the `list` command again shows that XGalaga is installed (denoted by the 'i' in column 2):

```
 i      xgalaga                    2.0.34-1                  Clone of the
classic game of galaga
```

An alternative to the command-line is FinkCommander, which provides a graphical user interface on top of Fink. See Figure 2. In addition to displaying package summary information, FinkCommander provides menu items that correspond to the Fink commands. You select the package to install or remove, then the Binary > Install or Binary > Remove command, respectively. Note that you must install Fink before installing FinkCommander.



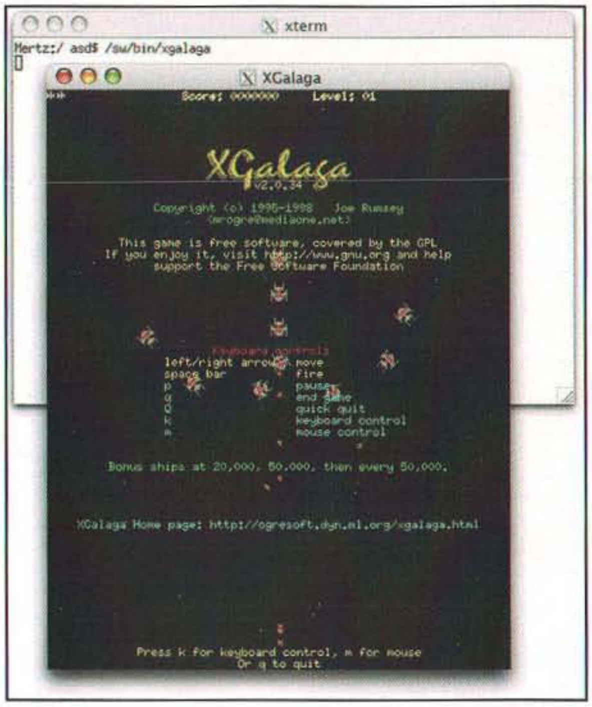
Status	Name	Installed	Latest	Binary	Category	Description
	xft2-shlibs		4.3.99.16-2			virtual package
current	xgalaga	2.0.34-1	2.0.34-1	2.0.34-1	games	Clone of the classic game of galaga
	xiangqi	2.0.34-1	2.0.34-1			virtual package
	ximian-conn		1.4.7-2		crypto	MS Exchange plugin for evolution
current	xinvaders	2.1.2-2	2.1.2-2	2.1.2-2	games	Space Invaders clone for X
	xinvaders3d		1.3.6-1	1.3.6-1	games	3D Space Invaders Game
current	xjig	2.4-12	2.4-12	2.4-12	games	Jigsaw puzzle game for X11

```
Reading Package Lists...
Building Dependency Tree...
The following NEW packages will be installed:
  xinvaders
0 packages upgraded, 1 newly installed, 0 to remove and 1  not upgraded.
Need to get 46.8kB of archives. After unpacking 131kB will be used.
Get:1 http://us.dl.sourceforge.net 10.3/release/main xinvaders 2.1.2-2 [46.8kB]
Fetched 46.8kB in 4s (10.6kB/s)
Selecting previously deselected package xinvaders.
(Reading database ... 4046 files and directories currently installed.)
Unpacking xinvaders (from .../xinvaders_2.1.2-2_darwin-powerpc.deb) ...
Setting up xinvaders (2.1.2-2) ...
```

Done

Figure 2: Using FinkCommander to Install a Package. The Fink Output Appears in the Text Area Beneath the Package List.

Now you can run the installed application. Figure 3 shows the command line used to launch XGalaga, and the application splash screen.



```
Mertz:/ asd$ /sw/bin/xgalaga
```

```
Score: 0000000     Level: 01

Copyright (c) 1995-1998   Joe Rumsey
(mrogre@mediaone.net)

This game is free software, covered by the GPL
If you enjoy it, visit http://www.gnu.org and help
support the Free Software Foundation

Keyboard controls
left/right arrow      move
space bar             fire
p                     pause
q                     end game
Q                     quick quit
k                     keyboard control
m                     mouse control

Bonus ships at 20,000, 50,000, then every 50,000.

XGalaga Home page: http://ogresoft.dyn.ml.org/xgalaga.html

Press k for keyboard control, m for mouse
Or q to quit
```

Figure 3: Running XGalaga in Shared Screen Mode

Installing and Running OpenOffice

OpenOffice is an open source office suite that aims to provide many of the features found in commercial Office software. This includes word processing, spreadsheet, presentation, and drawing capability. The link to the Mac OS X download of OpenOffice is provided at the end of this article. The OpenOffice download does not require Fink. However, the OpenOffice installer requires Mac OS X v10.3 Panther or later.

After downloading, run the installer. This will place an OpenOffice folder inside /Applications. Inside this folder, double-click `Start OpenOffice.org` (see Figure 4) to launch OpenOffice and the X11 environment (if not already running).

Internet. Local Phone. Long Distance

VOIP is Here!!

PHONEPIPE™

Unlimited Residential	$24.99
Unlimited Business	$49.99

www.PhonePipe.com

1-877-300-3035

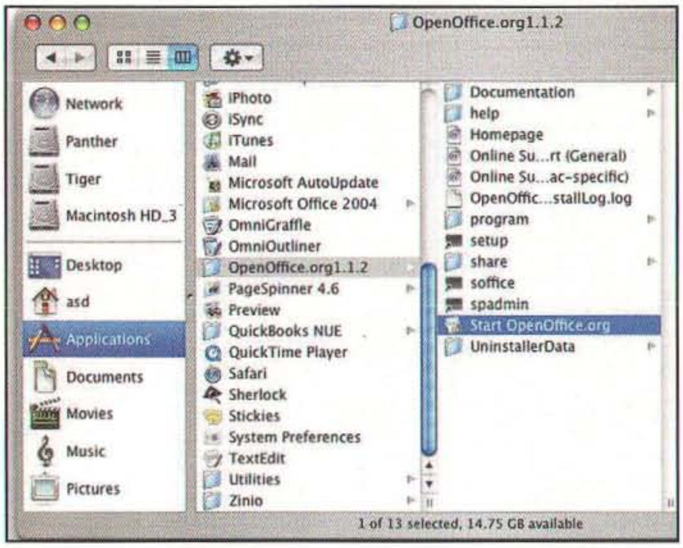
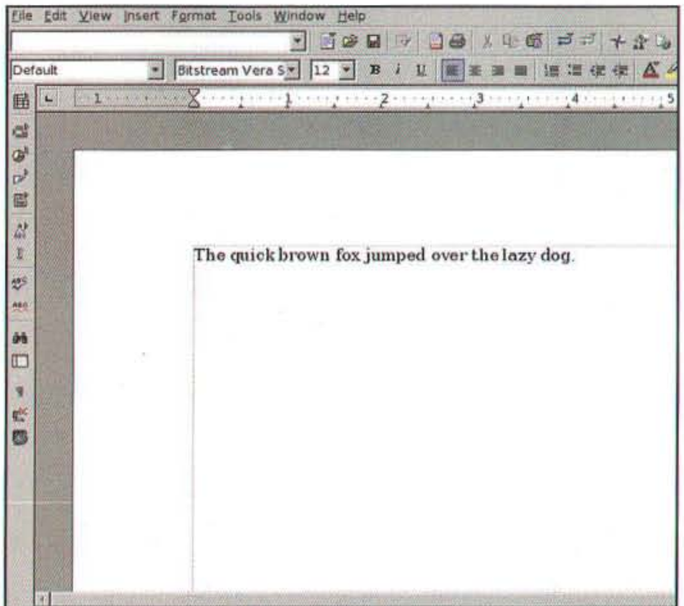


Figure 4: The Start OpenOffice Application

There is no additional setup required to run OpenOffice. The OpenOffice.org creators have made it very easy to get started. Remember that this is an X11 application, not an Aqua application, so it will look and behave differently than a native Mac OS X app. For example, text rendering in the OpenOffice word processor (see Figure 5) looks less sharp than in TextEdit, and the OpenOffice file formats are not necessarily compatible with their commercial counterparts. But OpenOffice offers a lot of functionality that makes up for these shortcomings.



The quick brown fox jumped over the lazy dog.

Figure 5: The OpenOffice Word Processor

Like other open source projects, OpenOffice will improve more quickly based on contributions made by the developer community. Check out the OpenOffice.org website if you would like to help out.

More Information

X11 can provide an additional source of software. The references below contain additional information on X11 and related topics.

* Several resources are linked off the Darwin X11 page (Darwin is Apple's open source projects).
http://developer.apple.com/darwin/projects/X11/index.html

* Technical Q&A QA1232 X11 FAQ.
http://developer.apple.com/qa/qa2001/qa1232.html

* Technical Q&A QA1383 Enabling X11 Forwarding. Additional information is available at Hacking Linux Exposed.
http://developer.apple.com/qa/qa2004/qa1383.html
http://www.hackinglinuxexposed.com/articles/20040705.html

* A great discussion of ssh -X versus xauth versus xhost is available on the Oroborus for Mac OS X site.
http://oroborosx.sourceforge.net/remotex.html

* Download X11 for Mac OS X (Note that users of Mac OS X Panther v.10.3 already have this download available on the CD, as well as the X11 SDK).
http://www.apple.com/macosx/features/x11/download/

* Download OpenOffice for Mac OS X (X11 version).
http://porting.openoffice.org/mac/ooo-osx_downloads.html

* Download Fink and FinkCommander from SourceForge.
http://fink.sourceforge.net/download
http://finkcommander.sourceforge.net/

* An overview of the X11 environment, including window managers and Fink, is provided in Mac OS X for Unix Geeks from O'Reilly & Associates.
http://www.oreilly.com/catalog/mpantherunix/

* For a popular and useful discussion, see Fink and Apple's X11.
http://homepage.mac.com/sao1/fink/index.html

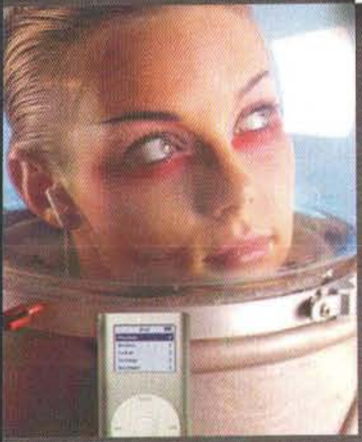
MT

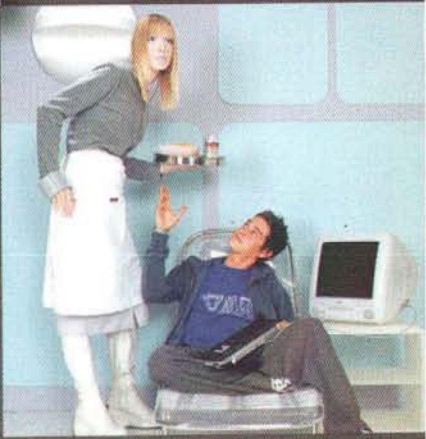
About The Author

Paul T. Ammann has been working in IT for almost 20 years now. He is happily married to his wife Eve for 6 years. He finds writing the author's bio the toughest part of the article.

$32 < 1 Year

2 Years > $64





Interviews

Tapping into the world of celebrities and their Macs, only MacDirectory offers exclusive interviews. Get a close and personal view from Sarah Jessica Parker, Steve Jobs, Madonna, Harry Connick Jr., George Lucas, Jennifer Jason Leigh, Steve Woz and other leaders in the Mac community.

Features

Designers, writers, musicians, business leaders & our technical expert team offer their own personal interpretation of things that only the Mac system can deliver. With more than 200 pages of news, insights, trends and the largest Macintosh buyer's guide including over 5,000 Mac products and services.

MacDirectory

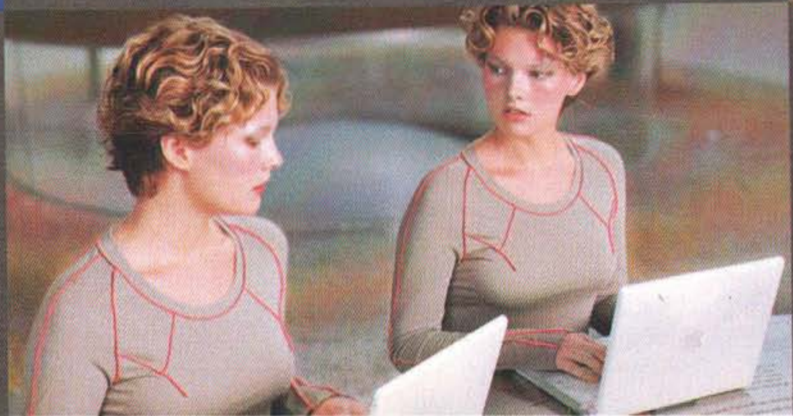
BEYOND ANY MACINTOSH MAGAZINE.

Culture

MacDirectory takes you to the wildest corners of the world and uncovers how Macintosh computers are being used by other cultures. Travel to Japan, Australia, Germany, Brazil & Russia and learn more about Apple's cultural impact around the globe.

Reviews

Find out all you need to know about the latest Mac products including the hottest Mac OS software and hardware.



Subscribe >

macdirectory.com

Send check or money order to:
MacDirectory Subscription Dept.
326 A Street, 2C
Boston, MA 02210

VODCASTING

EXPLORING THE FUTURE OF SYNDICATED MEDIA

Finding, Viewing and Creating Vodcasts on the Mac - Your 15 minutes of fame are coming right up!

By Emmanuel Stein

Editor's Note: Apple's recent announcement of native video support for iPod makes the future of Vodcasting even more certain... and more significant. Read on!!

Podcasting: The Next Generation of Radio? or Something more...

During his keynote address at WWDC 2005, Steve Jobs introduced podcasting as a time-shifting TIVOesque technology aimed at revolutionizing radio. Interestingly, no announcement was made of video support within the podcasting specification, not to mention iTunes playback of syndicated video feeds (e.g. vodcasts). Nevertheless, one is hard pressed to find more than a token sampling of extant vodcast content available for subscription via Apple's iTunes Music Store (ITMS). This strategy is consistent with Jobs' recent statement that "we also offer video podcasts, but will people buy a video device just to watch this video? So far they haven't. Nobody has been successful with that yet." It seems, at least for the time being, that Apple is content with offering basic support for vodcast content and letting the third party developers behind DTV and FireAnt innovate the medium in concert with the growing community of video bloggers. This situation is not likely to change until Apple develops a portable video solution, capable of higher quality video playback. On the content aggregation front, both DTV and FireAnt have been working with content providers and popular directory services like Del.icio.us, Yahoo, MeFeedia and many others to provide end users with a rich array of vodcast content.

Whether you call it vlogging, video blogging, Internet TV, or simply vodcasting (Video On Demand-casting), the idea behind podcasting, whether using video or audio, is certainly nothing revolutionary. Rather, podcasting is an evolutionary extension to the RSS 2.0 (Real Simple Syndication) standard, with support for media enclosures (e.g. audio, video, or pdf files). Nonetheless, Apple's announcement of podcasting and the related, though not well advertised, support for video enclosures certainly represents a shot in the arm for the burgeoning syndicated video crowd.

I would not be surprised if Apple were currently in talks with major media outlets to offer something like the ITMS, only with a focus on video content. However, at this point, Apple's strengths are their QuickTime 7 technology, with support for H.264 video compression, and their suite of video editing and content production applications like iMovie and Final Cut Pro. These tools will prove essential to Mac users in the pre-publishing phase of vodcast creation and will be used in the accompanying tutorial that covers turning your videos into vodcasts!

For the purposes of the article, the term vodcast will be treated as synonymous with more historical terms for the publishing of Internet-based video (vlogging, video blogging, vidcasting etc.), the bulk of which is vodcast compliant anyway. In tests, I have found that even RSS feeds made without reference to the podcasting specification work flawlessly with iTunes and integrate into the iTunes database as podcasts. The essential dependence, so far as iTunes is concerned, is RSS 2.0 compliance and iTunes support for your content type.

Although we will not weigh in on the debate surrounding nomenclature for this new medium, I would tend to agree with Michael Verdi, of http://www.michaelverdi.com/, whose rant on the subject (http://ia300026.us.archive.org/0/items/MichaelVerdiVlogAnarchy/vloganarchy.mov) is among the most lucid commentaries on this debate I have come across. His main thesis is that to define such a new and dynamic medium at this early stage would only serve to limit further experimentation and evolution within the medium. For my part, I couldn't agree more.

Anatomy of a Vodcast

Vodcasting is simply an instance of podcasting in which the target media is video. Therefore, the podcasting specification, as published by Apple at http://phobos.apple.com/static/iTunesRSS.html, is equally valid for several media formats (e.g. mp4a, mp3, mov, and pdf). If iTunes support is not required, you may use additional content formats like Flash, WMV, DivX, XviD, and others depending on which players you are targeting. Both DTV and FireAnt offer support for media formats beyond what is currently available in iTunes and employ plug-in architectures to facilitate the development of wider support for emerging and existing video formats.

Whichever approach you take to vodcasting, the steps remain the same:

1. Compress your video in one of many available formats depending on your playback solution of choice (e.g. iTunes, DTV, FireAnt, etc.)
2. Edit a simple XML file (iTunes compliant vodcast example provided in Listings 1 and 2) that details your vodcast metadata and may reference one or more media enclosures
3. Upload your XML and media files to a server, iDisk, or other hosting solution
4. Fire up iTunes, DTV or FireAnt and subscribe to your vodcast via reference to your XML file

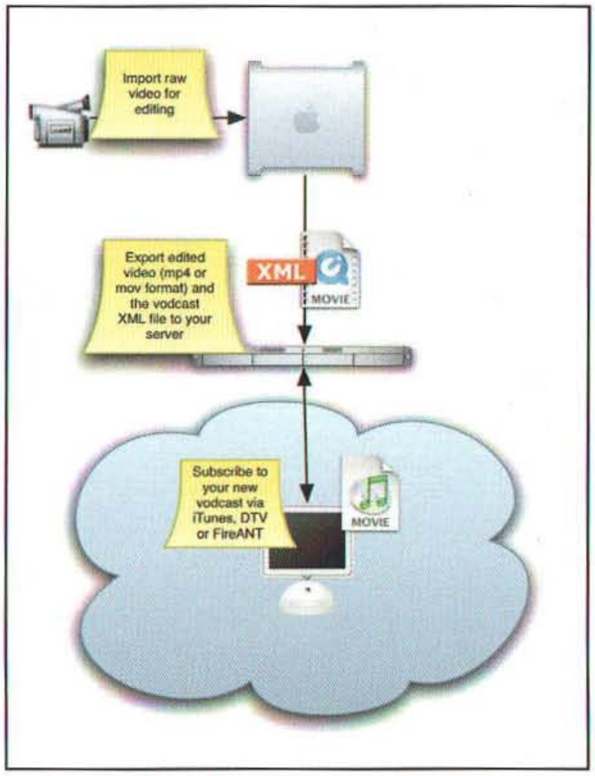
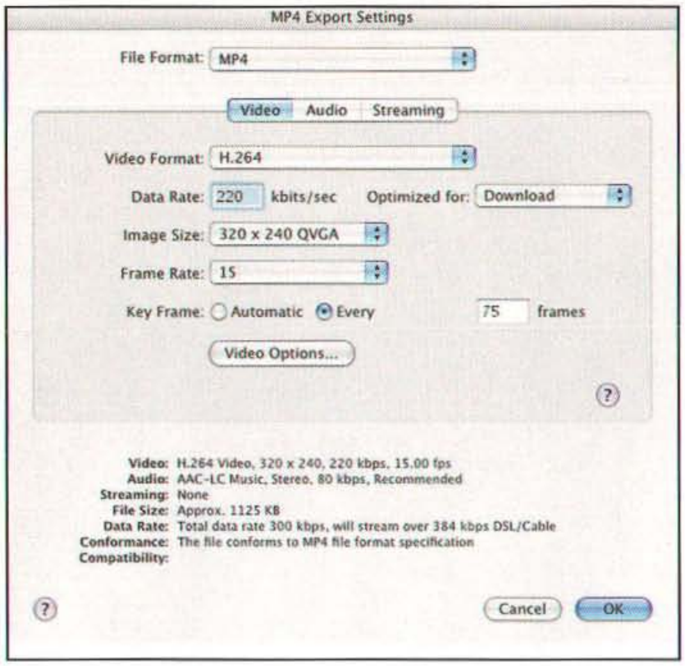

Import raw video for editing
Export edited video (mp4 or mov format) and the vodcast XML file to your server
XML
MOVIE
Subscribe to your new vodcast via iTunes, DTV or FireANT
MOVIE

Figure 1. A simple vodcast workflow diagram

Lights, Camera, Compression!

Once you have created a video, open it in QuickTime Pro and go to `File > Export` and select a video format compatible with iTunes and other players, such as mp4. Optimize the export by clicking on the `Options` button in the export window and tweaking the compression parameters in the `MP4 Export Setting` dialog (Figure 2). Feel free to experiment with the various settings or just go with the defaults. With playback support from alternative players like FireAnt, you can syndicate flash content like http://happytreefriends.atomfilms.com/index.html to great effect. However, if you want to retain iTunes 5.01 compatibility, you need to stick with either mov, or better still, mp4-based content. Depending on several factors (e.g. cpu speed, video format, data rate, image size, ***et alia)*** the export may take a few minutes or quite a bit longer.

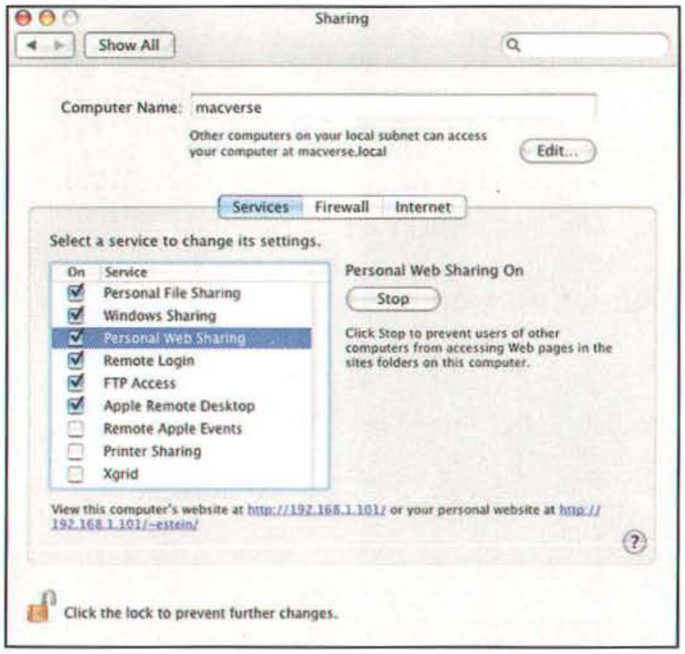


File Format: MP4
Video Format: H.264
Data Rate: 220 kbits/sec Optimized for: Download
Image Size: 320 x 240 QVGA
Frame Rate: 15
Key Frame: ○ Automatic ● Every 75 frames

Video: H.264 Video, 320 x 240, 220 kbps, 15.00 fps
Audio: AAC-LC Music, Stereo, 80 kbps, Recommended
Streaming: None
File Size: Approx. 1125 KB
Data Rate: Total data rate 300 kbps, will stream over 384 kbps DSL/Cable
Conformance: The file conforms to MP4 file format specification
Compatibility:

Figure 2. Sample settings for video compression using QuickTime 7 Pro with H.264

For this example, we will test the feed using your local web server. Therefore, you need to place the video and XML files into your `Sites` subfolder and enable `Personal Web Sharing` in the `Sharing` preferences pane (Figure 3).



Computer Name: macverse

Other computers on your local subnet can access your computer at macverse.local

Select a service to change its settings.

On	Service
✓	Personal File Sharing
✓	Windows Sharing
✓	Personal Web Sharing
✓	Remote Login
✓	FTP Access
✓	Apple Remote Desktop
☐	Remote Apple Events
☐	Printer Sharing
☐	Xgrid

Personal Web Sharing On

Click Stop to prevent users of other computers from accessing Web pages in the sites folders on this computer.

View this computer's website at http://192.168.1.101/ or your personal website at http://192.168.1.101/~estein/

Click the lock to prevent further changes.

Figure 3. Turn on Personal Web Sharing to host your feed on your local machine

Feed your Videos

For the sake of simplicity, we will stick with Apple's podcasting specification, which works fine with virtually all-available playback solutions. Although tools exist for creating vodcast feeds, I find it easier to just edit the file directly in a text editor (e.g. TextEdit or vi). Listing 1 and 2 provide you with a template upon which you may base your own vodcast XML file. Notice that the vodcast.xml file may be conceptually broken down into two main sections,. the first of which is populated with metadata that is consistent across vodcast episodes. In other words, this initial section remains the same for each instance of your vodcast, since, in principle, vodcasts refer to several media files. The second part of the file is specific to each edition of your vodcast as represented by a media enclosure. Therefore, the data used for Listing 2 will change for each added video enclosure you integrate within your feed.

In a perusal of the sample file, you will see that the tags are filled in with descriptive placeholder metadata that serves as an example of what each tag is meant to communicate. For a detailed discussion of tag definitions, please refer to http://phobos.apple.com/static/iTunesRSS.html. It is especially important to avoid illegal characters within your tags (e.g. &, >, <, ', ", ©, and ™) as they will render an otherwise well-structured feed unusable.

Listing 1: vodcast.xml (Section 1)

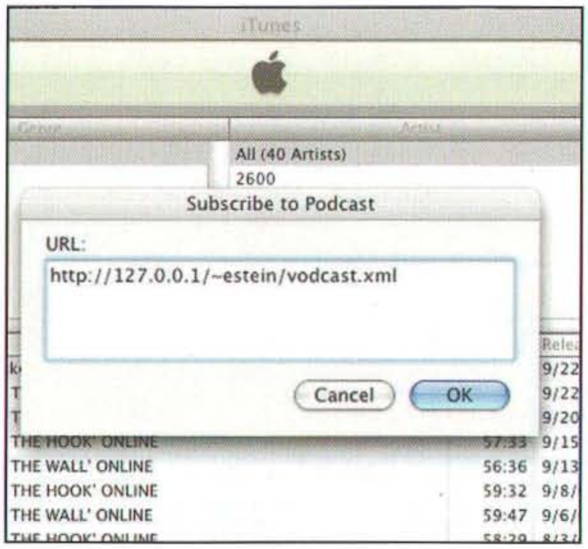
```
<?XML version="1.0" encoding="UTF-8"?>
<rss XMLns:itunes="http://www.itunes.com/DTDs/Podcast-1.0.dtd"
version="2.0">
<channel>
<title>MY VODCASTS</title>
<itunes:author>YOUR NAME</itunes:author>
<link>http://www.sitename.com</link>
<description>Put any description about the Vodcast channel you
desire here</description>
<itunes:subtitle>Put any subtitle about the Vodcast channel
you desire here</itunes:subtitle>
<itunes:summary>Put any summary about the Vodcast channel you
desire here</itunes:summary>
<language>EN</language>
<copyright>(c) 2005 your name</copyright>
<itunes:owner>
<itunes:name>yourname</itunes:name>
<itunes:email>youraddress@yourcompany.com</itunes:email>
</itunes:owner>
<category>Technology</category>
<itunes:category text="Technology"></itunes:category>
```

Listing 2: vodcast.xml (Section 2)

```
<item>
<title>MY FIRST MOVIE</title>
<itunes:author>yourname</itunes:author>
<description>Put any description about this first movie
here</description>
<itunes:subtitle> Put any subtitle about this first movie here
</itunes:subtitle>
<itunes:summary> Put any summary about this first movie here
</itunes:summary>
<enclosure url="http://www.yoursitename.com/moviename.mp4"
length="1024" type="video/mov" />
<guid>http://www.yoursitename.com/moviename.mp4</guid>
<pubDate>Sun, 24 July 2005 10:00:00 GMT</pubDate>
<itunes:explicit>no</itunes:explicit>
<itunes:duration>00:01:10</itunes:duration>
<itunes:keywords>keyword1, keyword2, keyword3</itunes:keywords>
</item>
</channel>
</rss>
```

Putting it all together

Once you have populated your XML file and enclosed a link to your video file, we can begin testing your vodcast. The main thing to remember is that your feed is represented by the XML file you created for your vodcast. Therefore, whether you are using iTunes, FireAnt, or another syndicated video player, all you need do is insert the URL pointing to your vodcast.xml file, which, if you are using the supplied template and Personal Web Sharing, should be http://127.0.0.1/~YourUserName/vodcast.xml. Using iTunes, you may test your vodcast by going to `Advanced > Subscribe to Podcast` and then entering your feed URL in the resulting dialogue box (Figure 4). The subscription process is almost identical for DTV and FireAnt, both of which have an `Add Channel` button allowing you to enter your feed URL.



Subscribe to Podcast

URL:

http://127.0.0.1/~estein/vodcast.xml

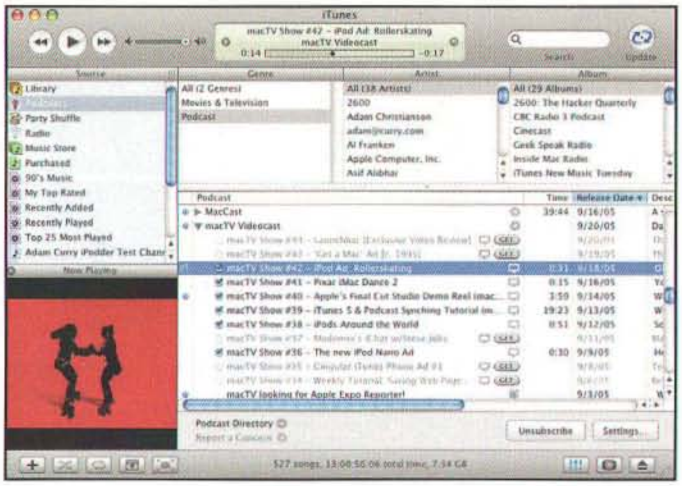
Cancel | OK

```
THE HOOK' ONLINE   57:33  9/15
THE WALL' ONLINE   56:36  9/13
THE HOOK' ONLINE   59:32  9/8/
THE WALL' ONLINE   59:47  9/6/
```

Figure 4. Subscribing to a locally hosted test vodcast in iTunes

DTV and FireAnt: Cures for the iTunes Vodcast Blues

Currently, iTunes (version 5.01) represents a poor viewing and browsing experience for the wealth of video feeds available. The ITMS also falls short with a fair, yet poorly organized sampling of syndicated video content, which one can only get to through extensive searching. What is more, iTunes version 5.01 plays vodcast content, by default, in the miniscule "album art" pane located in the lower left corner of the iTunes interface (Figure 5).



```
Podcast                                                   Time   Release Date
▶ MacCast                                                 39:44  9/16/05
▼ macTV Videocast                                                9/20/05
    macTV Show #42 – iPod Ad: Rollerskating               0:31   9/18/05
    macTV Show #41 – Pixar iMac Dance 2                   0:15   9/16/05
    macTV Show #40 – Apple's Final Cut Studio Demo Reel imac...  3:59  9/14/05
    macTV Show #39 – iTunes 5 & Podcast Synching Tutorial im...  19:23 9/13/05
    macTV Show #38 – iPods Around the World               0:51   9/12/05
    macTV Show #36 – The new iPod Nano Ad                 0:30   9/9/05
    macTV looking for Apple Expo Reporter!                       9/3/05
```

Figure 5. Default iTunes vodcast playback

To Apple's credit, you can click on the movie in the newly renamed song-artwork-video viewer to open a separate and resizable window (Figure 6). This solution is less than ideal inasmuch as the floating video playback window obscures the iTunes interface and is easily lost in the background. As you can imagine, this can be annoying and is entirely unnecessary as the music video playback model, seen in Figure 7, demonstrates that Apple has already developed a better playback solution. iTunes also has the option for full-screen playback, but this is not too well suited for shorter videos that tend to be highly compressed.

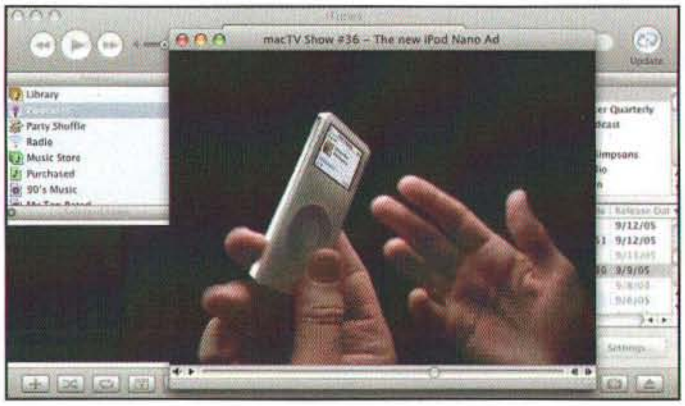


Figure 6. The iTunes disembodied playback window

BLOB XML PHP C++
REALbasic Director COM .NET
SQL Object-Relational Client Server Unicode
Joining Worlds of Information

PARADIGMA software

Valentina 2: Deploy True, Royalty Free Client-Server Solutions
www.paradigmasoft.com

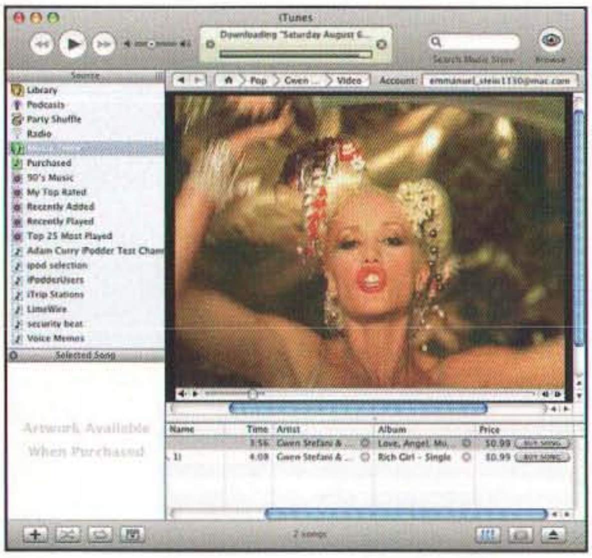


Figure 7. The way vodcast playback should be—Default ITMS music video playback.

Both DTV (http://participatoryculture.org/download.php) and FireAnt (http://getfireant.com/) present better alternatives for those wishing to get the full experience of vodcasting with accompanying directory and community resources. Both support a wider range of media formats relative to iTunes and also come with a nice selection of content or "Channels" to get you started.

FireAnt grew out of the video blogging community at Yahoo Groups (http://groups.yahoo.com/group/videoblogging/) and was developed in tandem with the nascent community of video bloggers. Furthermore, their solution was in the works before podcasting was adopted by iTunes. It is therefore not surprising that FireAnt has several unique community oriented features such as `Comment` and `Mail Post URL to a Friend`. Support for tag-based searching of directory content from Yahoo, FireAnt, Del.icio.us, and MeFeedia directories represents another unique feature of this application. FireAnt even offers the option of syncing your vodcasts with iTunes. Under the hood, FireAnt boasts a plug-in engine, allowing various additional media formats like Flash to be used as vodcast enclosures. The only drawback to FireAnt is that, unlike iTunes and DTV, you have only three sizing options (small, large and maximum) for playback and switching between them restarts the video being played. Otherwise, FireAnt offers a rich viewing and browsing platform for vodcast media and is a must have for anyone serious about vodcasting.

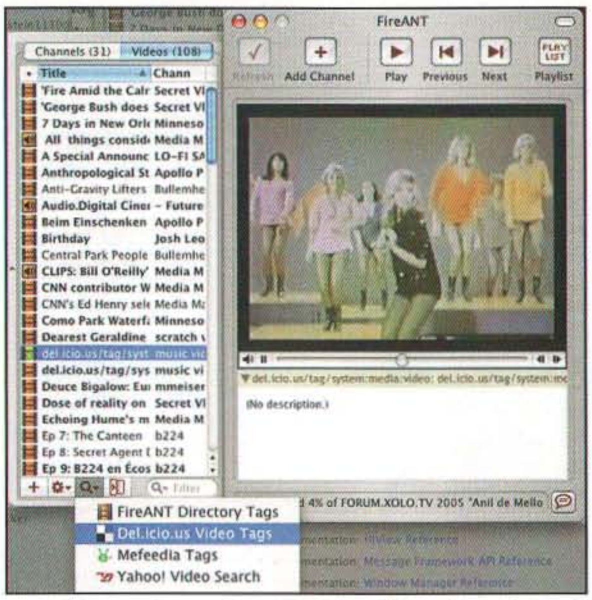


Figure 8. Nancy Sinatra does FireAnt with tag-based searching!

DTV is another excellent vodcast platform for the Mac and offers a more pared down, but very functional set of features relative

betterRAM.com℠

Memory built to order - so you always get the right memory with up-to-the-minute prices!

Toll Free: (800) 895-3493 • Outside US/Canada: 805-494-9797 • Fax: 805-494-9798

LOOK FAMILIAR?

Get Rid of Your Fax Machine!

Receive your faxes directly to your email account as PDF file attachments

Service available in over 150 area codes

To check out special offers for MacTech readers, visit www.MaxEmail.com/MacTech

maxemail®

to FireAnt. First off, tag-based searching and community-oriented functionality are not available as of this writing. Nonetheless, DTV has a simple and intuitive interface that allows for real-time video resizing. As an open source project, DTV benefits from the rapid and robust development cycle associated with this development paradigm. Further, Participatory Culture Foundation, the organization behind DTV, offers a server-based Broadcast Machine (http://participatoryculture.org/bm/)–a scalable solution for both individuals and larger workgroups looking to automate or otherwise streamline their vodcast production. Future versions of DTV will be based on VLC, a great open source video player that comes with all the codecs needed to view the wide range of vodcast content.

Content is King

Many of today's popular content producers like Steve Garfield, of http://stevegarfield.blogs.com/, were pioneering Internet-based video media as early as 1997! Steve's Vlog Soup is highly recommended as an introduction to the community of video bloggers. Using a format similar to Comedy Central's Talk Soup, Vlog Soup offers hilarious excerpts of various extant video blogs with equally humorous commentary by Steve. Another star of the indie vodcast scene is RocketBoom (http://www.rocketboom.com/vlog/), a daily news program with correspondents all over the globe, which features classic comedic performances by host Amanda Congdon. XOLO TV (http://www.xolo.tv/) is another news-style vodcast, with particularly compelling interviews done by Gabe, the host of the 'cast. Mac-Heads (myself included!) will enjoy the MacTV vodcast (http://live.watchmactv.com/) with content ranging from OS X software tutorials to techno-lusty advertisements of Apple products.

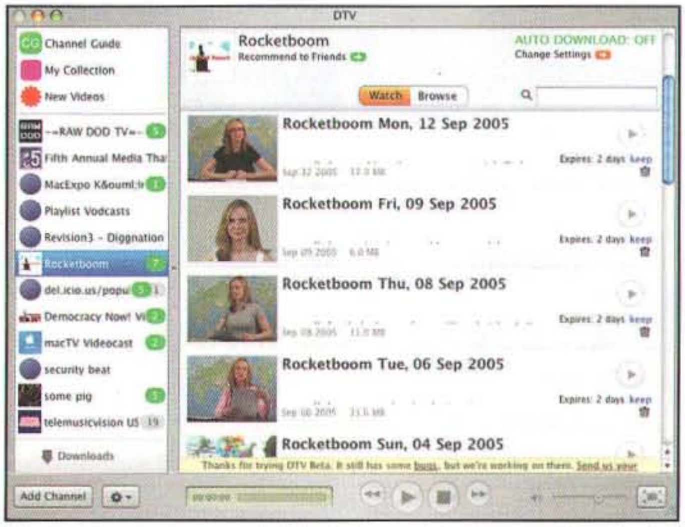


Figure 9. Previewing RocketBoom episodes in DTV

Vodcasting and associated syndicated video technologies are nothing, in and of themselves, if people don't use them. There is a wide range of video content available and just wadding through it can be quite a job in itself. Not to worry, though, since both DTV, FireAnt and to a lesser extent the ITMS, offer an aggregation of content to get you started. However, to really get a sense of what is out there, you will want to explore the wide variety of directory services and community oriented forums.

Directories

http://del.icio.us/

Del.icio.us bills itself as a social bookmarks manager and is one of the most popular directory services for vodcasting content. It allows users to create a personal link collection and add metadata to characterize their content. Also available are social features that enable you to browse the collections of other users and otherwise share information. Further, their API (http://del.icio.us/doc/api) allows developers to better integrate their content with del.icio.us specific tags as demonstrated in FireAnt (Figure 8).

http://www.mefeedia.com/

With the claim of being the first video aggregator, Mefeedia certainly has lots to offer with a robust set of social features, custom tags for content and instructional information. What is more, they even help you host your vodcast, all for free. Like Del.icio.us, Mefeedia's tag-based searching is integrated into FireAnt and is available to developers.

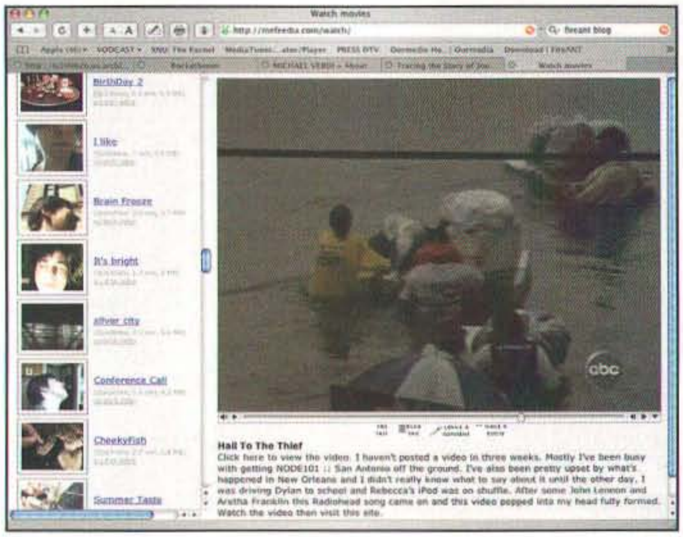


Figure 10. Surveying the damage of hurricane Katrina as I browse popular vodcasts at Mefeedia.com

http://video.search.yahoo.com/

Yahoo has long played host to many in the vodcasting or video blogging community and, not surprisingly, offers a capacious selection of content. Additionally, Yahoo offers a tagging system, which like the last two directories detailed, is integrated with players like FireAnt. Perhaps more interesting, is the Yahoo Groups community for video bloggers that will be detailed in the community links section to follow.

http://vlogdir.com/

This directory offers a nice selection of vodcasts with some of the social elements that have become so popular among video blog directory services.

http://getfireant.com/directory.php

One of the most active groups developing solutions for vodcasters, FireAnt also offers a tagging system and directory service that has drag and drop subscribing functionality with their player.

Community, Hosting and Instructional Sites

http://groups.yahoo.com/group/videoblogging/

This is one of the oldest and largest online communities of vodcasters on the Net and is definitely worth a look. FireAnt, as well as, many of today's top bloggers got their start here.

http://blip.tv/

Not unlike some of the directory services covered, blip.tv allows for sharing and viewing of vodcasts, along with free hosting.

http://videoblogging.info/

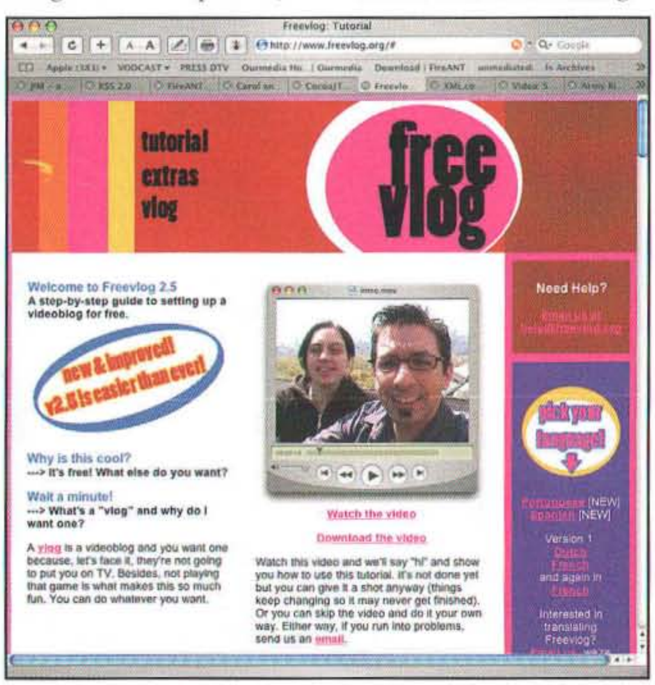
This site offers unique content such as a world map of video bloggers, in addition to instructional articles, support, and an array of useful community links. If you need a hosting solution, look elsewhere as this site does not offer any such services. However, the site does supply useful information gleaned from the major video blogging communities and is therefore worth a look.

http://www.ourmedia.org/

Ourmedia is a great site for obtaining all types of content, including vodcasts, and represents another hosting solution for sharing your videos and other media. Further, links to tutorials and support are also available. Ourmedia is a great overall solution for users wishing to explore, create and publish multi-media content and share it with a growing community of users.

http://www.freevlog.org/

A wonderful and fun resource, freevlog.org represents the heart and soul of the vodcasting community with its rich array of instructional material and links to free hosting space to boot. If you plan on doing any vodcasting, I highly encourage you to visit this site as it really does a wonderful job in guiding you through the whole process, from content creation to hosting.



Welcome to Freevlog 2.5
A step-by-step guide to setting up a videoblog for free.

Why is this cool?
---> It's free! What else do you want?

Wait a minute!
---> What's a "vlog" and why do I want one?

A vlog is a videoblog and you want one because, let's face it, they're not going to put you on TV. Besides, not playing that game is what makes this so much fun. You can do whatever you want.

Watch this video and we'll say "hi" and show you how to use this tutorial. It's not done yet but you can give it a shot anyway (things keep changing so it may never get finished). Or you can skip the video and do it your own way. Either way, if you run into problems, send us an email.

Portuguese [NEW]
Spanish [NEW]
Version 1
Dutch
French
and again in
French
Interested in translating Freevlog?

Figure 11. Freevlog will help you to become a vodcast kung-fu master!

Say you want a Revolution...

Whether you call it vodcasting or video blogging, there are certainly no signs of this new medium and paradigm for media syndication ebbing anytime soon—quite to the contrary. Expect iTunes to further integrate vodcasts into their feature set and provide additional categories in the ITMS, so as to offer better browsing and differentiation of vodcast content from audio podcasts. Given the fact that Apple has recently updated their iPod patent filing to add coverage of a video player, it is likely that, in the secrecy characteristic of the new Apple, Jobs and company are working on a video iPod and an accompanying content aggregation and distribution scheme, possibly with larger and more traditional broadcast companies.

Either way, it is certain that vodcasting will increasingly blur the lines between TV and Internet-based media. With a trend toward more niche content, we may indeed see a phenomenon akin to the explosion of print news outlets in the 1900s (between 1910-1914 the number of U.S. periodicals peaked with 2,600 dailies and 14,000 weeklies!), but with a 21st Century twist. Unlike traditional broadcast media, vodcasters tend to aggregate themselves into a community and use the medium not only for production of mainstream content, but also for interaction with friends and family. This type of personalized approach is refreshing and a boon in an age of globalization with little social and inter-group contact. With the barrier to entry quickly receding, alternative syndicated media technologies, like vodcasts, will be well poised to compete with "mainstream" media as an interactive populist force that may well overtake or force a major paradigm shift among the old guard of today's media conglomerates. With little differentiation among the major content providers, the chorus of fresh new voices among the independent vodcasting set is sounding sweeter by the minute!

MT

About The Author

Emmanuel Stein has been an avid Mac user since 1984 and has honed his cross-platform skills while working at France Telecom, Time Magazine and Reed-Elsevier. He has recently started his own Mac-centric consulting company, MacVerse, which offers implementation, system administration and development services geared towards the enterprise market. As a diehard GNU/Linux geek, he enjoys hacking open source software and experimenting with new open source projects on OS X. You may reach him at macverse@mac.com

PROTECT YOUR POD

These lightweight super-tough aluminum cases for the iPod® are the most durable and stylish EVER!

Customized engraving and other colors available.

www.A1QP.com

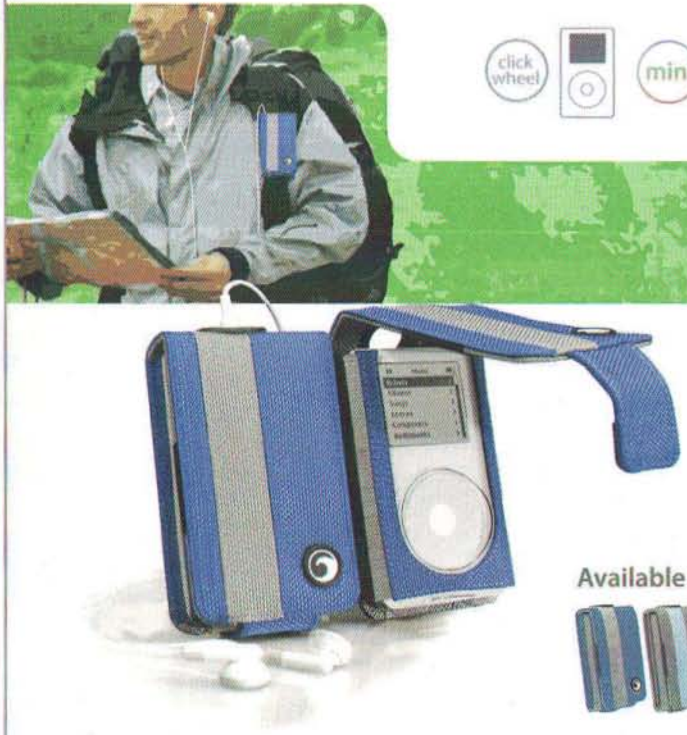
Designed and manufactured in the USA by A-1 Quality Products, Inc.
©2005 All rights reserved. iArmor and iKeychain are trademarks of A-1 Quality Products, Inc.
iPod is a registered trademark of Apple Computer, Inc.

What's the ***difference***?

Brand new. It's perfect. It's beautiful...	Brand new. It's perfect. It's beautiful...
and it's *vulnerable*!	and it's **protected**!
It's naked, exposed to the elements.	Invisible, durable, & rugged protection.
It's... INHUMANE and it ***will*** get scratched.	SAVE the iPod

www.the***invisible*****SHIELD**.com

Trail Vue™



Includes:
iPod case
Removable belt clip
Fits all click wheel models

A versatile case

You didn't compromise on your music player—
don't compromise on your case.

Available colors

www.MARWARE.com

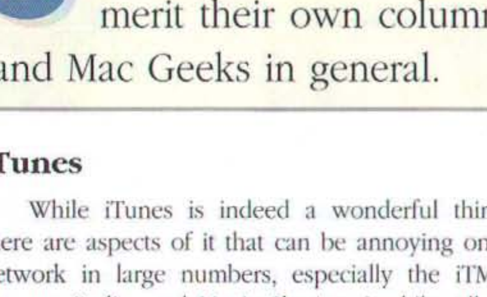
COLLECTIONS AND CONTEMPLATIONS

IT TYPES FINALLY GET A ROOM OF OUR OWN

So after the last series we saw here, I thought that I'd devote this column to a collection of items that aren't enough to merit their own columns, but still of use to Mac IT Admins and Mac Geeks in general.

iTunes

While iTunes is indeed a wonderful thing, there are aspects of it that can be annoying on a network in large numbers, especially the iTMS, Internet Radio, and Music Sharing. Luckily, all of these can be managed, some of them centrally. With iTunes 5, you can now set preferences for accessing Podcasts, the iTMS, and Shared Music from within the "Parental" section of the iTunes preference. Yes, I know that doesn't cover Internet radio, but if you look in the "General" section, you see the control for showing or displaying Internet Radio. Finally, you have further controls in the "Sharing" section. These sections are shown in figures 1a – 1c below.



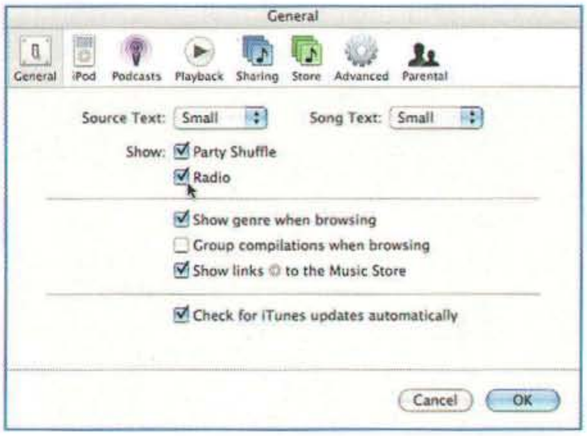
Sources: ☐ Disable Podcasts
☐ Disable Music Store
☐ Disable shared music
Music Store: ☐ Restrict explicit content
Click the lock to prevent further changes.

Figure 1a: iTunes 5.X Parental Controls

However, you still have to get those preferences out to the user's machines. For that, you really, ***really*** want to use Workgroup Manager and the MCX (Managed Client for OS X) capabilities that allow you to push out individual preference files. I would go into the details of how to use this with iTunes, but as it turns out, I don't have to. Instead, go to the .Mac site of John DeTroye, Apple SE and MCX wizard extraordinaire. Go to his downloads section, select "Latest_Tips", "Tiger-tips" and download the "mini-tandt-itunes5.pdf" document. It will show you how to use Workgroup Manager to manage iTunes for everything but Internet Radio.

Now, that's not a minor issue. Internet Radio is potentially a huge bandwidth hog, especially if you multiply each connection by a couple hundred or thousand users. However, there's two ways to deal with this, one elegant, but requiring a more advanced firewall, and one that's not so advanced, but works well nonetheless.

The elegant way is to block the initial request from iTunes. As it turns out, iTunes makes all its initial requests for things like Internet radio and the iTMS as http connections. In those connections, it has a user agent, that, on my machine shows up as: ***User-Agent: iTunes/5.0.1 (Macintosh; N; PPC)***. So, if your firewall/router setup is able to handle higher level filtering, you simply tell it to block all outbound HTTP traffic where the User-Agent contains "iTunes". That blocks all versions, all platforms. At that point, your iTunes traffic is now local – only. *Note: While*



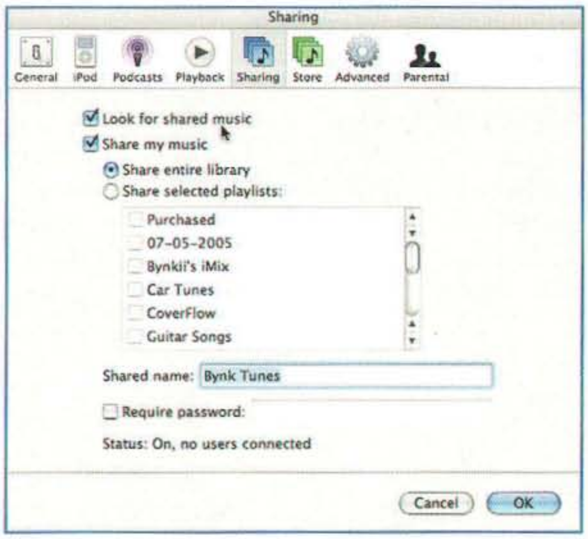
Source Text: Small   Song Text: Small

Show: ☑ Party Shuffle
☑ Radio

☑ Show genre when browsing
☐ Group compilations when browsing
☑ Show links to the Music Store

☑ Check for iTunes updates automatically

Figure 1b: iTunes General Controls



☑ Look for shared music
☑ Share my music
(●) Share entire library
( ) Share selected playlists:
☐ Purchased
☐ 07-05-2005
☐ Bynkii's iMix
☐ Car Tunes
☐ CoverFlow
☐ Guitar Songs

Shared name: Bynk Tunes

☐ Require password:

Status: On, no users connected

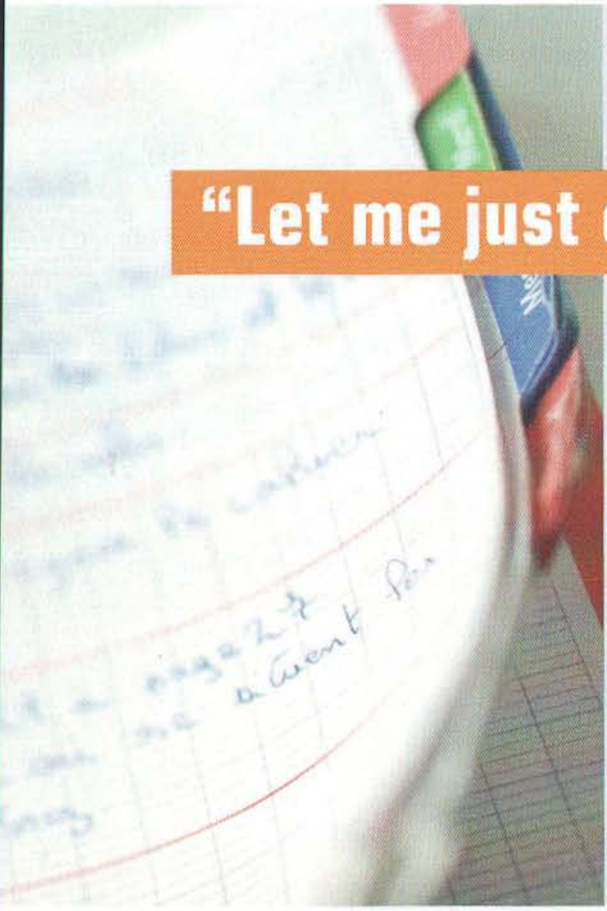
Figure 1c: iTunes Sharing Controls

there are a lot of very complex tools to discover things like this, my personal favorite here is tcpflow, available via DarwinPorts at http://www.darwinports.org/.

If your firewall/router setup isn't able to do this, then there's a simpler, albeit uglier way. Block TCP ports 8000-8999 and 42000-42999. That will prevent any iTunes Internet Radio streams to your network. Doing the iTMS is a little trickier, since that all happens over ports 80 and 443, and if you block those, you've effectively cut off the World Wide Web. However, if you kill access to "phobos.apple.com", you can block off the iTMS, at least until Apple changes the DNS name of the iTMS.

To block music sharing, (say if you don't yet have all your machines on iTunes 5.x yet), just set the firewalls on the individual Macs to block all connections on TCP port 3689. (This can be done any number of ways, from shell to Applescript; the specific implementation is really up to your individual

Connect • Communicate • Collaborate • Securely



"Let me just check my calendar..."

Kerio MailServer

A groupware alternative to Exchange that syncs calendars, contacts and email with Entourage and Outlook. Integrated anti-spam and McAfee virus filtering provide secure, junk-free email for users on any platform.

Mac OS X · Linux · Windows

www.kerio.com

McAfee SECURITY

KERIO

preferences and skillset. You can also do it as part of the imaging process for new machines, and let attrition handle it for you.) If for no other reason, the fact that iTunes 5 lets you block everything but Internet Radio with relative ease, is a good reason to upgrade.

AppleScript Tricks

So, as many who know me can attest to, my .sig file in Entourage is huge and varied. However, I got rather tired at typing them in manually all the time, so...AppleScript to the rescue. I have two scripts that handle signature creation, one from within emails in Entourage, the other for things I see in Safari that are theft-worthy.

The Entourage script is fairly simple:

```
set theSigTitleRecord to display dialog "Enter a name for
       the signature" default answer "RandomSig 1"

set theSigTitle to text returned of theSigTitleRecord

tell application "Microsoft Entourage"
       try
              set theSelection to the selection as text
              set theSelection to "— " & return &
       theSelection
              make new signature with properties
                     {name:theSigTitle,
                     content:theSelection, include in
                     random:true}
       end try
end tell
```

The first part is easy. We display a dialog that asks for a name to be used for the signature, with some default text. Dialogs all return a record, so we grab the "text returned" field of that record, and put it in theSigTitle.

The rest all happens within Entourage. We get the selected text, drop it into theSelection, and make sure it's plain text. We then set up the sig format in theSelection, which by RFC is "—<space><return><sigtext>". With Entourage, you need to use the return keyword, not the \r escape for returns. We then create a new signature with the required properties; name, content, and is it in the random list, (yes). By wrapping it in a try block, I deal with any errors. There should be some error checking to look for me trying to run this script without selected text, but since I'm the only one (until now), who's using it, it hasn't been a big deal.

The Safari version is similar:

```
set noSelectedTextFlag to 0

tell application "Safari"
       set theText to (do JavaScript
"getSelection()" in document 1)
       if theText = "" then --some sites with
              frames don't allow for the
              JavaScript above, so copying is the
              fix
       tell application "Safari" to activate
              tell application "System Events"
                     tell process "Safari"
                            keystroke "c" using
                     {command down}
                            delay 0.5
                            set theText to the
                     clipboard
                     end tell
              end tell
       end if
       if theText = "" then
              display dialog "You need to have
       something selected!"
              set noSelectedTextFlag to 1
       end if
end tell

if noSelectedTextFlag = 0 then
       set theSigTitleRecord to display dialog
              "Enter a name for the signature"
              default answer "RandomSig 1"
       set theSigTitle to text returned of
theSigTitleRecord

       tell application "Microsoft Entourage"
              try
                     set theText to "— " & return
              & theText
                     make new signature with
                            properties
                            {name:theSigTitle,
                            content:theText,
                            include in
                            random:true}
              end try
       end tell
end if
```

As we can see, the script only has a few changes. The first line is a flag for some error checking that I added to the script, and defaults to 0. In the Safari section, we first try to use JavaScript within Safari to get the selected text and put it in theText. If that doesn't work, and it often doesn't, we then resort to the quick 'n' dirty UI scripting method, and have Safari act as though we hit cmd-C to get the selected text onto the clipboard. That is then dumped into theText.

The next line is a quick error check. If, after all that, theText is still empty, then we display a dialog informing the user that hey, this won't work so well without actual selected text, and it sets noSelectedText to 1. From there, we check to see if noSelectedText is 0. If it is, then we create the signature. If not, then we don't and the script ends.

True, neither of these scripts are all that complex or "work – oriented" but they do give you some ideas of how to accomplish the same thing from two different angles, and some very basic introduction to using JavaScript and UI scripting in the same AppleScript. Besides, its fun to have a large collection of pithy signatures.

Microsoft Office 2004 Service Pack 2

While there are always arguments for and against applying a service pack or update, if you use Entourage in a Microsoft Exchange environment, run, don't walk to apply this. It has a host of fixes and changes for Exchange users that people have been asking about for some time. It doesn't do everything everyone wanted, but it hits a lot of issues like delegation, folder sharing, password change messages, sync speed, GAL usage, quota management, and Public Folders right out of the park.

As well, the Entourage Weblog, at http://blogs.msdn.com/entourage/default.aspx is no longer dormant, and has a bunch of really great articles about Entourage's SP2 changes. There are some fixes to the rest of Office, but after all, Entourage is why we really buy Office, right? (I'm so getting in trouble for that ;-)

Conclusion

Again, nothing major here, just some small "storylets" that I've had bouncing about for a bit, and decided to turn into a column. Sometimes, you just have to go light.

Bibliography and References

Much thanks to John DeTroye for his tips and tricks documents, they're a boon to Mac administrators everywhere. http://homepage.mac.com/johnd

The folks at MacSurfer originally created the Safari code to make their lives easier, it works really well, so I stole it like a thief in the night. MacSurfer is also the best news aggregation page on the Mac web, and I hit it at least 5 times a day. http://www.macsurfer.com/.

We may not always like their parent company, but the Microsoft Macintosh Business Unit always does great work, and I can't imagine trying to work without Entourage, Word, and PowerPoint. http://www.microsoft.com/mac/.

MT

About The Author

John Welch <jwelch@bynkii.com> is the Unix/Open Systems Admin for Kansas City Life Insurance, a Technical Strategist for Provar, (http://www.provar.com/) and the Chief Know-It-All for TackyShirt, (http://www.tackyshirt.com/). He has over fifteen years of experience at making Macs work with other computer systems. John specializes in figuring out ways in which to make the Mac do what nobody thinks it can, showing that the Mac is a superior administrative platform, and teaching others how to use it in interesting, if sometimes frightening ways. He also does things that don't involve computertry on occasion, or at least that's the rumor.

Website Not Found By Clients
HTTP 404 - Website not found

The website you are looking for can't be found by your clients. It may have been improperly marketed, had poor design, or didn't work. Regardless, it's not helping your business!

Your Options:

- Rent a chicken suit and stand on the corner handing out flyers
- Paint the company URL on your chest and face during a major sporting event
- Contact SharpNET Solutions internet marketing and web design specialists, watch your traffic and rankings increase, get great feedback from all your new customers.

HTTP 404 - Website needs SharpNET

Not Marketing Your Site?
If you are not marketing your website online, you may as well have a 404 error for a website.

Web Design
Internet Marketing
Consulting
Lead Generation
Multi-Media

1-877-583-8396
www.sharpnetsolutions.com

INTRODUCTION TO SCRIPTING ICAL

For the past couple of articles, I have discussed scripting specific applications. By now, you should be starting to realize AppleScript terminology varies from application to application. Some applications don't support AppleScript at all, some are more scriptable than others, some have more confusing terminology, etc. Even as become more knowledgeable as a scripter, you will find that there is a learning curve whenever you need to script a new application. Browsing the application's dictionary, and any accompanying documentation or example scripts is usually your best bet for learning how to script a new application.

This month, I will continue to discuss application-specific scripting, and this time, I will focus on iCal. Please note that all sample code within this article was written and tested with Mac OS X Tiger 10.4.x. Many times, software updates will introduce changes in the AppleScript terminology of a given application or process. Therefore, if you are using an older system, some code may not work properly, or may need to be adjusted slightly to work on your machine.

Working with Calendars

Let's begin by discussing the highest-level class within iCal's object hierarchy, a calendar. In iCal, a calendar contains events and to do's, which we will discuss a little later. For now, let's talk about creating calendars, locating them, and more.

Making a New Calendar

More than likely, you are working with existing calendars within iCal, such as a "Home" or "Work" calendar. However, should you need to, you do have the ability to create calendars via AppleScript.

iCal actually contains a command specifically for creating calendars, called simply enough `create calendar`. This command has an optional parameter, which will allow you to specify the calendar's name. The following sample code demonstrates the creation of a calendar with the use of this command.

```
tell application "iCal"
  create calendar with name "My Calendar"
end tell
```

One thing to take note of when using this command is that it does not produce a result. Because of this, you cannot place the newly created calendar into a variable for future reference. As an alternative to the `create calendar` command, you can also use the `make` command. This may be a better choice in this situation, as the `make` command does produce a result, that result being the newly created item. For example, the following code demonstrates how the `make` command can be used to create a calendar. In this example, the result of this command – the newly created calendar – is placed into a variable called `theCalendar`, which I can now reference later in my script.

```
tell application "iCal"
  set theCalendar to make new calendar with
```

```
properties {title:"My Calendar"}
end tell
--> calendar 3 of application "iCal"
```

Getting a List of Calendars

If you are working with existing calendars, you may need to write code that will retrieve a list of these calendars. The following code demonstrates how to retrieve a list of all existing calendars.

```
tell application "iCal"
  set theCalendars to every calendar
end tell
--> {calendar 1 of application "iCal", calendar 2 of
application "iCal", calendar 3 of application
"iCal"}
```

The previous code will return references to your existing calendars. However you , may only need to retrieve the names of your calendars. To do this, simply retrieve the title of every calendar. For example:

```
tell application "iCal"
  set theCalendarNames to title of every calendar
end tell
--> {"Home", "Work", "My Calendar"}
```

Changing Calendar Views

Within iCal, you have the option to view calendars in several different ways. You can use scripting to change calendar views as well. The following example code demonstrates how to change the calendar view to display in day view mode.

```
tell application "iCal"
  switch view to day view
end tell
```

The following sample code demonstrates how to change the calendar view to display in week view mode.

```
tell application "iCal"
  switch view to week view
end tell
```

The following example code demonstrates how to change the calendar view to display in month view mode.

```
tell application "iCal"
  switch view to month view
end tell
```

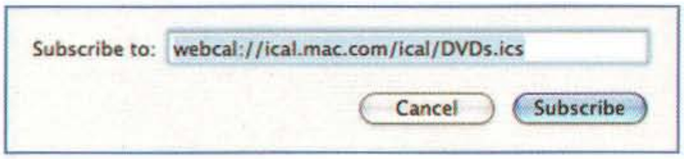
So, with the use of the code above, you can actually customize the viewing experience of iCal for the user, as your script processes.

Subscribing to a Calendar

In addition to creating calendars, you can also write code to subscribe to a calendar, using a calendar URL. The following code demonstrates how this is done.

```
tell application "iCal"
  GetURL "webcal://ical.mac.com/ical/DVDs.ics"
end tell
```

Please note that, when subscribing to a calendar via AppleScript, some manual intervention will be required. iCal will pop up a few dialogs, allowing users to confirm that they actually do want to subscribe to the calendar, and also allowing them to configure subscription settings. See figure 1.



webcal://ical.mac.com/ical/DVDs.ics

Figure 1. Calendar Subscription Dialog

Working with Events and To Do's

Now that we have discussed calendars, let's talk about elements of calendars. Specifically, we will walk through several tasks involving events and to do's.

Making a New Event

Like calendars, events can be created with AppleScript. To do this, use the make command. When creating an event, you will most likely want to specify values for various attributes of that event. For example, for an event, you may want to specify the event's name, description, location, etc. This can be done as the event is created, with the use of the with properties parameter. The following sample code demonstrates how to create a new all-day event for the current day, with a specified name, description, and location.

```
tell application "iCal"
  tell calendar "My Calendar"
    set theDate to current date
    make new event at end with properties
{description:"Event Description", summary:"Event
Name", location:"Event Location", start date:theDate,
allday event:true}
  end tell
end tell
--> event 1 of calendar 3 of application "iCal"
```

To create an event that falls within a specific time period, you may specify the start date and end date properties of the event. For example, the following example code will create an event at the current date and time, with a length of 2 hours.

```
tell application "iCal"
  tell calendar "My Calendar"
    set theCurrentDate to current date
    make new event at end with properties
{description:"Event Description", summary:"Event
Name", location:"Event Location", start
date:theCurrentDate, end date:theCurrentDate + 120 *
minutes}
  end tell
end tell
--> event 1 of calendar 3 of application "iCal"
```

Making a New To Do

The process of creating a to do is virtually the same as that of creating an event. Again, you can use the `make` command, and you may optionally specify properties to be applied as the to do is created. The following sample code will create a to do with a specified summary, description, and due date.

```
tell application "iCal"
  tell calendar "My Calendar"
    set theDueDate to (current date) + 30 * days
    make new todo at end with properties
{description:"To Do Description", summary:"To Do
Name", due date:theDueDate}
  end tell
end tell
--> todo 1 of calendar 3 of application "iCal"
```

To find a complete list of event and to do properties, consult the appropriate class in the ***iCal*** suite in iCal's AppleScript dictionary.

Finding an Event or To Do

Many times, you may be working with existing events or to dos. If this is the case, then you might need to locate the appropriate event or to do in some way. The best way to locate something in iCal is to do so by its unique ID. In iCal, calendars, events, and to do's all have unique ID's, which can be retrieved by AppleScript from the item's `uid` property. The following sample code demonstrates how to retrieve the ID of an event.

```
tell application "iCal"
  tell calendar "My Calendar"
    set theEvent to first event
    return uid of theEvent
  end tell
end tell
--> "1BCA3512-F3A9-4BCB-A0FD-BE812968D371"
```

If you have the ID of an event or to do, you can then find the item by its ID. The following code shows how to locate an event by its unique ID.

```
tell application "iCal"
  tell calendar "My Calendar"
    set theEvent to first event whose uid =
"1BCA3512-F3A9-4BCB-A0FD-BE812968D371"
  end tell
end tell
--> event 1 of calendar 3 of application "iCal"
```

This same technique may be used to locate a to do by its unique ID.

Viewing an Event or To Do

AppleScript can also be used to display a specific event or to do within iCal. To do this, use the `show` command, and specify the item that you want to display. For example, the following sample code will cause iCal to display and select a specific event, which, in this case, is stored in a variable named `theEvent`.

```
tell application "iCal"
  tell calendar "My Calendar"
    show theEvent
  end tell
end tell
```

Deleting an Event or To Do

Just as you can create events and to do's in iCal, you can also delete them. To delete an item in iCal, use the `delete` command and specify the item that you want to delete. For example, the following code will delete a specified event that is stored in a variable.

```
tell application "iCal"
  tell calendar "My Calendar"
    delete theEvent
  end tell
end tell
```

Working with Alarms

If you are an avid iCal user, then you are probably already aware that events and to do's can be configured with alarms. There are multiple types of alarms that you can configure manually, which can also be configured via scripting. AppleScript can be used to create the following types of alarms:

- **Display Alarm** – This type of alarm will display a message to the user, letting the user know about a scheduled event or to do. With the use of scripting, you can set the trigger interval or date for this type of alarm.
- **Mail Alarm** – This type of alarm will send an email message to the current user, notifying the user of an upcoming event or to do. Like a display alarm, the trigger date or interval for this type of alarm may be set via scripting.
- **Open File Alarm** – This type of alarm will open a file at a specified time. AppleScript can be used to set the date or interval for the alarm. It can also be used to specify the path to the file that should be opened. An alarm of this nature can be extremely useful if you want a script to trigger at a specific time.
- **Sound Alarm** – This type of alarm will produce an audio alert about an upcoming event or to do. For this type of event, AppleScript may be used to specify the date or interval, as well as the name or file path of a sound to be used for the alert.

Adding an Alarm to an Event or To Do

Let's take a moment to look at how an alarm can be created with the use of AppleScript, for a given event or to do. For this first example, we are going to create a display alarm. The following example code

MACTECH®
The Journal of Macintosh Technology

Now for all Mac experts!
See the new *MacTech Magazine*

Are you a power user? Like getting under the hood? Join the crowd. Find out what's causing all the buzz in the technical community. Check out RISK FREE the new MacTech: Now for all techs!

As a 20 year old publication, the All-New MacTech has been featuring an improved design and a greatly expanded editorial focus. Each and every month in MacTech you'll find in-depth coverage and how-to articles. Unlike many magazines that tell you what you *can* do, MacTech shows you *how* to make the most of your Mac experience.

The NEW MacTech Magazine is for anyone that wants to know what's under the hood. Whether a power user, hobbyist, IT Pro, or consultant MacTech has something to offer you.

MacTech is already read every month by tens of thousands of readers, from power users to IT Pros to developers. They represent the very heart and soul of the Macintosh technical community. Join the most influential part of the community, and sign up for MacTech today!

Be the one with all the answers. Make your Mac fly. Get under the hood.

For a one year subscription visit:

store.mactech.com/riskfree

MACTECH
Tiger Server At Your Service
By John Welch
REALbasic 2005 - What's New
DNS And E-mail
QuickTime: Developing Applications with The QT Kit - Part III
Journaling: A Look At The OS X Personal Diary
Extending Spotlight's Reach
Building A Cocoa Screen Saver
Great Content That Is Thrill And Amaze You!
User Interface Scripting
Life With SQLite
launchd: Judge, Jury, Executioner
The Heart & Soul Of Open Source: Kenbatman and Source Hound
Fun With Quartz Composer: How To Build An App With No Code Whatsoever!
With Great, Coverage!
What's New With Applescript In Tiger
Access Control Lists: The Differences Between Access And Permissions

will add a display alarm to a specified event.

```
tell application "iCal"
  tell calendar "My Calendar"
    set theEvent to event 1
    tell theEvent
      make new display alarm at end with properties
{trigger interval:-30}
    end tell
  end tell
end tell
--> display alarm 1 of event 1 of calendar 3 of
application "iCal
```

In the code above, you will notice that the trigger interval property of the event is set to -30. The trigger interval property may be specified to configure when the alarm message should be displayed. This property should be given a numeric value, signifying minutes. In this case, I have specified a negative value of 30. This will cause the alarm to trigger 30 minutes prior to the start date of the event.

Optionally, I can choose to specify a trigger date for the alarm, rather than a trigger interval. This will allow me to configure the alarm to trigger at a specific date and time, rather than on a trigger interval. The following sample code will create a new display alarm for a given event.

```
tell application "iCal"
  tell calendar "My Calendar"
    set theEvent to event 1
    set theDate to (current date) - 3 * days
    tell theEvent
      make new display alarm at end with properties
{trigger date:theDate}
    end tell
  end tell
end tell
--> display alarm 1 of event 1 of calendar 3 of
application "iCal"
```

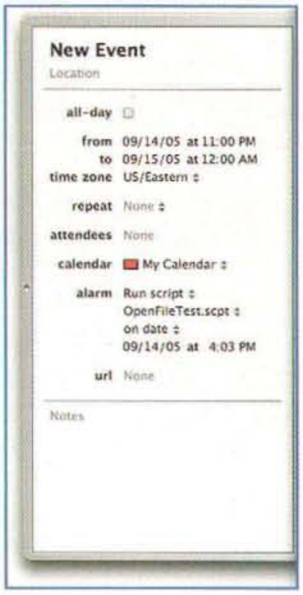
Let's look at another type of alarm. This time, let's add an open file alarm to an event. For this type of alarm, in addition to specifying a trigger interval or date, we can specify a file path for the item to be opened. The following example code demonstrates the process of creating an open file alarm for a given event.

```
set theFile to choose file
tell application "iCal"
  tell calendar "My Calendar"
    set theEvent to event 1
    set theDate to (current date) - 3 * days
    tell theEvent
      make new open file alarm at end with
properties {trigger date:theDate, filepath:theFile}
    end tell
  end tell
end tell
--> open file alarm 1 of event 1 of calendar 3 of
application "iCal"
```

If you look in iCal's dictionary, you may notice that the `filepath` property of an open file alarm is defined as needing a POSIX style path. As you can see from the code above, if you pass an AppleScript alias reference, that will work as well. To pass a POSIX path, you may need to convert the desired file's path. For example:

```
set theFile to POSIX path of (choose file)
--> "/Users/bwaldie/Desktop/FileToOpen.scpt"
```

Please note that if you configure an open file alarm to open a compiled AppleScript file, the script will actually be loaded and run by iCal, rather than simply opened. See figure 2. This is a great way to create a workflow with scripts that trigger at scheduled intervals.



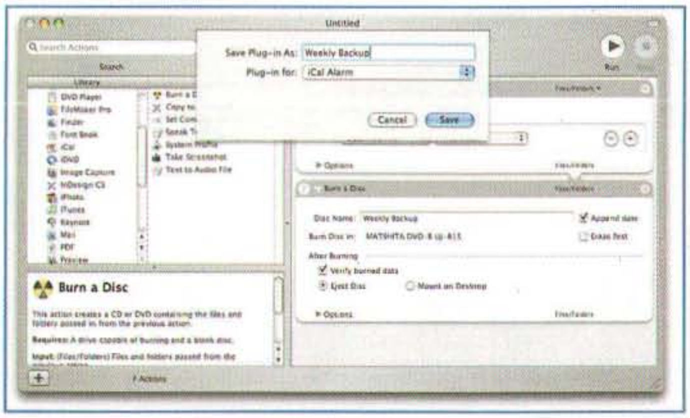
New Event
Location
all-day ☐
from 09/14/05 at 11:00 PM
to 09/15/05 at 12:00 AM
time zone US/Eastern
repeat None
attendees None
calendar 🟥 My Calendar
alarm Run script
OpenFileTest.scpt
on date
09/14/05 at 4:03 PM
url None
Notes

Figure 2. An Open File Alarm to Trigger a Compiled Script

Triggering Scripts and Automator Workflows from iCal

As we have seen above, there are ways to trigger an AppleScript from within iCal. You can quickly and easily create events, and add open file alarms to run compiled scripts, open script applications, etc.

You can also trigger Automator workflows from iCal, and Automator makes the process of configuring such scheduled events a snap. First, begin by creating an Automator workflow to perform any set of desired tasks. Next, select ***Save as Plug-In...*** from the ***File*** menu within Automator. From the plug-in type popup menu, select ***iCal Alarm***. See figure 3.



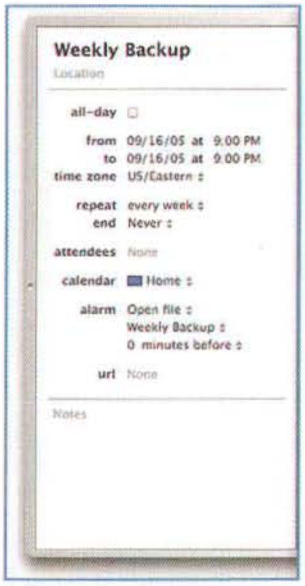
Save Plug-in As: Weekly Backup
Plug-in for: iCal Alarm

Disc Name: Weekly Backup ✓ Append date
Burn Disc in: MATSHITA DVD-R UJ-815 ☐ Erase first
After Burning
✓ Verify burned data
● Eject Disc ○ Mount on Desktop

Burn a Disc
This action creates a CD or DVD containing the files and folders passed in from the previous action.
Requires: A drive capable of burning and a blank disc.
Input: (Files/Folders) Files and folders passed from the

Figure 3. Creating an Automator iCal Alarm Plug-In

Click the ***Save*** button, and a new event will be created in iCal, which will be configured to trigger the workflow. Now, you can adjust the event, as needed, perhaps putting it on a repeating schedule. See figure 4 for an example of a configured event.



```
Weekly Backup
Location

   all-day  ☐

      from  09/16/05 at  9:00 PM
        to  09/16/05 at  9:00 PM
 time zone  US/Eastern ⇕

    repeat  every week ⇕
       end  Never ⇕

 attendees  None

  calendar  ▆ Home ⇕

     alarm  Open file ⇕
            Weekly Backup ⇕
            0 minutes before ⇕

       url  None

Notes
```

Figure 4. A Configured Automator iCal Alarm Plug-In Event

In Closing

Hopefully, you are already thinking of the great possibilities for creating scripts that interact with iCal, or that work in conjunction with iCal. In addition to creating and interacting with calendars, events, and to do's, you can also begin to schedule the execution of your scripts, allowing you to really begin putting your computer to work for you, perhaps at night, over the weekend, or whenever you are away from your desk.

For a list of some existing iCal scripts, try searching for "iCal" in the ScriptBuilders section of MacScripter.net at http://scriptbuilders.net/. For more information about creating and scheduling Automator workflows, check out the help files that come with Automator, or check out my Automator book, available from SpiderWorks at http://www.spiderworks.com/ books/automator.php.

Until next time, keep scripting!

MT

About The Author

Ben Waldie is author of the best selling books "AppleScripting the Finder" and the "Mac OS X Technology Guide to Automator", available from http://www.spiderworks.com. Ben is also president of Automated Workflows, LLC, a firm specializing in AppleScript and workflow automation consulting. For years, Ben has developed professional AppleScript-based solutions for businesses including Adobe, Apple, NASA, PC World, and TV Guide. For more information about Ben, please visit http://www.automatedworkflows.com, or email Ben at applescriptguru@mac.com.

BMS

THE LAW OFFICE OF BRADLEY M. SNIDERMAN

Need help safeguarding your software?

If you're developing software, you need your valuable work protected with trademark and copyright registration, as well as Non Disclosure Agreements.

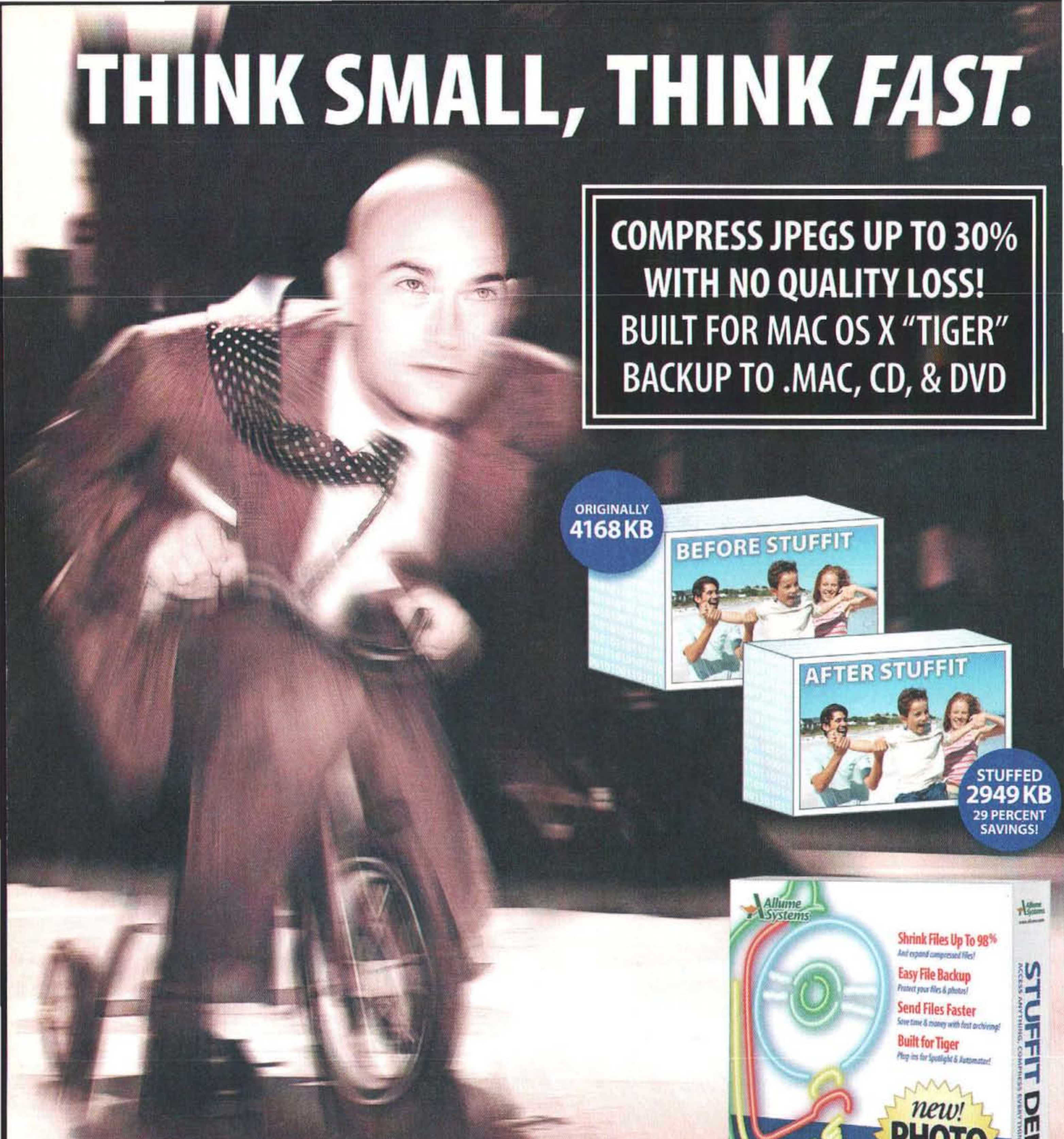
Then, when you are ready to sell it, you can protect yourself further with a licensing agreement.

I am an attorney practicing in Intellectual Property, Business Formations, Corporate, Commercial and Contract law.

Please give me a call or an e-mail. Reasonable fees.

23679 Calabasas Rd. #558 • Calabasas, CA 91302

PHONE 818-222-0365 FAX 818-591-1038 EMAIL brad@sniderman.com


THINK SMALL, THINK *FAST.*
COMPRESS JPEGS UP TO 30% WITH NO QUALITY LOSS!
BUILT FOR MAC OS X "TIGER"
BACKUP TO .MAC, CD, & DVD
ORIGINALLY 4168 KB
BEFORE STUFFIT
AFTER STUFFIT
STUFFED 2949 KB 29 PERCENT SAVINGS!
Allume Systems
Shrink Files Up To 98% And expand compressed files!
Easy File Backup Protect your files & photos!
Send Files Faster Save time & money with fast archiving!
Built for Tiger Plug-ins for Spotlight & Automator!
new! PHOTO JPEG COMPRESSION
STUFFIT DELUXE 10.0
ACCESS ANYTHING, COMPRESS EVERYTHING – EVEN PHOTOS!

Allume Systems

A Division of Smith Micro Software

Available from your favorite retailers and catalogs. Download StuffIt Expander for FREE at www.stuffit.com

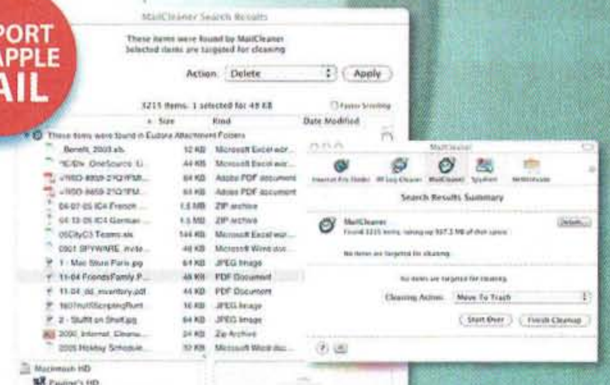
Download a FREE trial version of StuffIt Standard Edition at **www.allume.com.** For information about StuffIt's industry-leading JPEG compression, please visit **www.stuffit.com.**

®, TM, and © 2005 Allume Systems, Inc. Mac OS X 10.4 "Tiger" is required for StuffIt Deluxe's Spotlight and Automator support.

KNOW WHO'S WATCHING YOU.

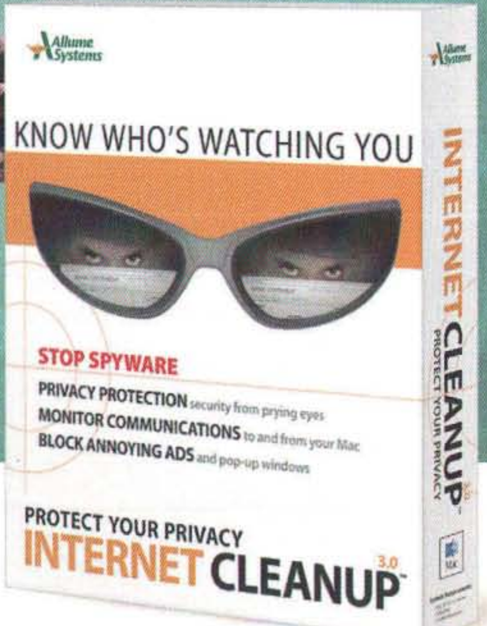
PROTECT YOUR PRIVACY
MONITOR COMMUNICATIONS
SECURELY ERASE FILES
BLOCK ANNOYING WEB ADS

SUPPORT FOR APPLE MAIL



```
MailCleaner Search Results
These items were found by MailCleaner
Selected items are targeted for cleaning
Action: Delete    Apply
3215 items. 1 selected for 48 KB       Faster Scrolling
      Size    Kind                 Date Modified
These items were found in Eudora Attachment Folders
Benefit 2003.xls            12 KB   Microsoft Excel wor...
*E/Div OneSource U          44 KB   Microsoft Excel wor...
VISO-8859-2?Q?FM...         64 KB   Adobe PDF document
VISO-8859-2?Q?FM...         63 KB   Adobe PDF document
04-07-05 IC4 French ...     1.5 MB  ZIP archive
04-13-05 IC4 German ...     1.5 MB  ZIP archive
05CityC3 Teams.xls          144 KB  Microsoft Excel wor...
0901 SPYWARE invite...      48 KB   Microsoft Word doc...
1 - Mac Store Paris.jpg     64 KB   JPEG Image
11-04 FriendsFamily.P...    48 KB   PDF Document
11-04 dd inventory.pdf      44 KB   PDF Document
1607nutRScriptingRunt       16 KB   JPEG Image
2 - Stuff on Shelf.jpg      64 KB   JPEG Image
2000 Internet Cleanu...     24 KB   Zip Archive
2005 Holiday Schedule...    32 KB   Microsoft Word doc...
Macintosh HD
Pauline's HD
Downloads/Attachments
11-04 FriendsFamily PDF.pdf
Show Preview

MailCleaner
Search Results Summary
MailCleaner
Found 3215 items, taking up 937.3 MB of disk space
No items are targeted for cleaning.
No items are targeted for cleaning.
Cleaning Action: Move To Trash
Start Over   Finish Cleanup
```



Allume Systems
A Division of Smith Micro Software

Available from your favorite retailers and catalogs. Download a FREE Trial at Allume.com!

One hundred percent 10.4 "Tiger" compatible! Download a FREE trial version of Internet Cleanup at **http://www.allume.com/mac/cleanup/** today!

Introduction to Core Data, Part III

Fetch, Clarus, Fetch

By Jeff LaMarche

In the previous two Core Data articles, we discussed how to create a data model and how to create, delete, and edit data instances both directly from the user interface and programmatically. Those two articles covered most of what you'll need to do with your application data models. Most. But not all. There's one important area that those articles didn't touch on.

In the prior articles, all instances of data that we worked with were ones that we created, or else were ones that the user took some action upon, so we never had to worry about what piece of data we were going to work with. There are times, however, when you will need to retrieve a piece of data from your application's context, but instead of knowing, for example, that the user clicked on the row that corresponds to a specific data instance, you'll know only some criteria about that data. For example, you might need to find a book instance based on its title or, perhaps, to find all the books by a particular author or publisher, or to find all the books published in a given year.

Now, certainly, you could create an `NSArrayController` in Interface Builder to represent all the instances of a particular entity type and then manually loop through the items it manages to look for the instances that meet your criteria, but that would be inefficient and involve a lot of unnecessary work. You're much better off using the tools Apple gives you for this purpose.

Fetch Requests

The mechanism that you use to retrieve data programmatically from your application's or document's context is called (appropriately) a *Fetch Request*, and is represented by the Cocoa class `NSFetchRequest`. To use this class, you instantiate a request object then provide it with the criteria that describes the object or objects that you want to retrieve. After that, you *execute* the fetch request, and it returns to you an array of `NSManagedObjects` that match those criteria. Fetch requests exist in both Enterprise Objects Framework (EOF) and in Core Data, and function very similarly, so if you've used EOF fetch requests or another object-relational mapping tool like Hibernate or Cayenne, this process will seem familiar to you.

Enterprise Objects has a very handy class of utility methods called `EOUtilities` which encapsulates a lot of commonly used functionality into public static methods (Java's equivalent to Objective-C's class methods) that can be called from anywhere. Core data has no comparable class, despite using a nearly identical mechanism for retrieving data.

So, this month, instead of building a whole new application, we're going to implement a class similar to `EOUtilities` (but much less extensive) called `MTCDUtilities` (that stands for MacTech Core Data Utilities, if you were wondering). The methods we are going to implement will make it easier to retrieve data from your context and also, in the process, will show you how to find and retrieve data from your application's context based on criteria. MTCDUtilities will have four class methods (much less than EOUtilities) and no instance variables, so you will never instantiate an MTCDUtilities object, but rather simply call methods on the class object.

The complete class we're building is available on the MacTech FTP site, with complete HeaderDoc documentation. Before we learn how to specify criteria for retrieving data, let's take a quick look at retrieving data without criteria. When you do not specify any criteria, a fetch request retrieves all the existing instances of a given entity. The first of our four convenience methods does exactly that. Before we write it, let's create our class' header, with declarations of the methods we're going to be writing.

MTCDUtilities.h

This is the header for MTCDUtilities. There are no instance variables, just four class method declarations. To save space, I have not included the HeaderDoc comments.

```
#import <Cocoa/Cocoa.h>
#define MTTooManyEntitiesReturnedException
  @"Too many entities returned"
#define MTCoreDataExeption @"Core Data exception"
@interface MTCDUtilities : NSObject
{
}
+(NSArray *)objectsForEntityNamed:(NSString *)name
  inContext:(NSManagedObjectContext *)context;
+(NSArray *)objectsForEntityNamed:(NSString *)name
  matchingKey:(NSString *)key
  andValue:(id)value
  inContext:(NSManagedObjectContext *)context;
+(NSManagedObject *)objectForEntityNamed:(NSString *)name
  matchingKey:(NSString *)key andValue:(id)value
  inContext:(NSManagedObjectContext *)context;
+(NSArray *)objectsForEntityNamed:(NSString *)name
  matchingKeysAndValues:(NSDictionary *)keyValues
  usingOR:(BOOL)useOR
  inContext:(NSManagedObjectContext *)context;
@end
```

For now, don't worry about any of the methods except for objectsForEntityNamed:inContext:, which we'll be writing first. We'll write the rest after we look at how to specify criteria.

MTCDUtilities.m / objectsForEntityNamed:inContext:

This method returns all instances of a given entity.

```
+(NSArray *)objectsForEntityNamed:(NSString *)name
  inContext:(NSManagedObjectContext *)context
{

  // We have to tell the Fetch Request which entity instances to
retrieve. We do
  // that using an NSEntityDescription based on the parameter name.
  NSEntityDescription *entity = [NSEntityDescription
    entityForName:name inManagedObjectContext:context];

  // Next, we declare an NSFetchRequest, and give it the entity
description
  NSFetchRequest *req = [[NSFetchRequest alloc] init];
[req setEntity:entity];

  // Before executing the fetch request, we declare an NSError
  // which is used to find out about any problems encountered
  // executing the request

  NSError *error = nil;

  // We next supply it (by reference) when we execute the request

 NSArray *array = [context executeFetchRequest:req
    error:&error];

  // A nil array tells us something went wrong

if (array == nil)
  {
    // We instantiate an exception using the error
description from
    // NSError, then raise the exception, which stops
execution

NSException *exception = [NSException
       exceptionWithName:MTCoreDataExeption
       reason:[error localizedDescription]
       userInfo:nil];

    // Since execution will stop when we raise the exception, we
    // need to release any memory before we do so.

    [req release];
    [exception raise];
  }

  // We allocated it, we have to release it

[req release];

  // Return the result set returned from executing the fetch request

  return array;
}
```

Now, any time we want to retrieve all the instances of a given entity, we can simply do something like:

```
NSArray *array = [MTCDUtilities
  objectsForEntityNamed:@"Book"
  inContext:context];
```

Predicates and Format Strings

That's all well and good, but when you want execute a fetch request to retrieve less than all of the instances, you're going to need to tell the request exactly which data instances you want to retrieve. The way that criteria are specified in Core Data is by way of something called a *predicate*, which is represented by the class NSPredicate and its subclasses.

An example of a predicate, in plain English, is "Name is 'Joe'" or "Age is less than 21". Predicates can be more complex, however, such as "Name is 'Joe' or 'Fred' or begins with the letter "I" and age is less than 21 or parent has given permission." The latter is an example of a compound predicate. Predicates are completely distinct from entities and fetch requests. You can create one predicate, say, "Name is 'Joe'" and use it to retrieve all the People entities with a name of Joe and then turn around and use the same predicate to retrieve all the Dog entities named Joe. Predicates don't know or care one whit about the entities they are being used to retrieve.

Predicates can be assembled programmatically by creating expressions, arguments, and substitution variables and compounding them into predicates. You will rarely, if ever, do this, however, because Apple has provided a much nicer mechanism for creating predicates: the Format String.

You've used format strings in Cocoa before. NSString has a similar mechanism that allows you to assemble multiple strings, raw datatypes, and Objective-C object instances into a single NSString using stringWithFormat:. The format strings used by NSPredicate are similar, but not exactly the same, as those used by NSString. Both types of format strings use substitution variables, but NSPredicate adds a bit of functionality. Because the format strings used to specify criteria has its roots in SQL (Structured Query Language—the language used to retrieve data from databases), internally it is fairly picky about certain language constructs. For example, if you're comparing a string attribute to a constant value, the constant value must be contained within single quotes.

For the most part, however, if you use format strings to create your predicates, you won't have to worry about such things. When you use NSPredicate's predicateWithFormat: method, it will automatically do the right thing based on the type of object you pass in as a substitution variable. If you give it a string, it will add the appropriate quotes. If you give it a date, it will translate it into the appropriate date string for comparison. If you give it an NSNumber, it will leave the quotes off..

Despite the fact that it is very savvy about dealing with substitution variables, NSPredicate is still pickier about format strings than is NSString. There are a limited number of operators that you can use to create a valid predicate. You can use all the major C and SQL operators that you are accustomed to, such as == or = for an equals comparison, > for greater than, < for less-than, != or <> for not equal to, >= for greater than or equal to, and <= for less than or equal to. You can also use AND (or &&), OR (or ||), or NOT (or !) to compound phrases in your format string.

Here are a few examples of creating a predicate using a format string:

```
NSPredicate *pred1 =
  [NSPredicate predicateWithFormat:@"age <= 21"];
NSPredicate *pred2 = [NSPredicate predicateWithFormat:
  @"name == %@ OR name == %@", @"Mary", @"Joe"];
```

Note that in the first example, I'm using a constant value (21) right in the format string. You can do this with numbers safely, but not with strings. You could not, for example, do this:

```
NSPredicate *pred = [NSPredicate
  predicateWithFormat:@"name == Bob"];
```

Why? Well, remember what I said earlier about NSPredicate being picky internally about things like quotes around strings? If you don't use substitution variables, you're responsible for making sure that your constant matches NSPredicate's internal format requirements. Since attribute names cannot be made up of solely numbers, there's no chance of confusion between a number constant and an attribute name or reserved keyword. The same isn't true for string constants.

In general, it's safer to always use substitution variables, even when you're using a constant value. If you're curious, the following code *will* work (notice the single quotes), though I don't recommend you make a habit of creating predicates this way:

```
NSPredicate *pred = [NSPredicate
  predicateWithFormat:@"name == 'Bob'"];
```

In addition to the operators mentioned above, there are also a slew of operators specifically for comparing string attributes, such as BEGINSWITH, CONTAINS, ENDSWITH, LIKE, and MATCHES. The first three are self-explanatory. The LIKE operator will be familiar to anyone who has worked with SQL: It functions identically to == except that it allows the use of two wildcard characters. When using LIKE, the character * works as an unbounded wildcard, so for example, 'w*t' would match 'wet', 'woot', and 'well I'd like to see you again if you permit it'. The ? character, on the other hand, is a bounded, single character wildcard, so 'w?t' would match 'wet' and 'wat' but not 'woot'.

MATCHES is similar to LIKE, except that it allows the use of full regular expressions rather than the more simplistic wildcards supported by LIKE. Regular expressions are beyond the scope of this article, so we won't be discussing MATCHES at all, but I will reiterate my comment from the first Core Data article that taking the time to learn regular expressions is well worth your time as a Cocoa programmer or OS X power user.

String operators have two optional modifiers (c and d) that can be specified in square brackets immediately following the operator. These can be used to specify (respectively) case sensitivity and diacritical sensitivity. By default, string comparisons in Core Data are both case-insensitive and diacritical-insensitive, meaning that if you create a predicate 'name CONTAINS bob', you would get back data instances where the attribute name equals 'Bob', 'bob', 'BOB', or even 'BÔB'. Here is an example of creating a predicate string that is case and diacritical sensitive or, in other words, here's how you get 'Bob' without getting his German cousin 'BÔB' or his Swedish cousin 'Bøb':

```
NSPredicate *pred = [NSPredicate predicateWithFormat:
  @"name LIKE[cd] %@", @"Bob"];
```

There are a few more operators available to you, but the ones I've mentioned will make up the vast, vast majority of what you'll use; you can feel free to investigate the others in Apple's Predicate documentation.

There's one more difference between the format strings used by NSString and those used by NSPredicate that needs to be mentioned: NSPredicate supports only two substitution variables, one of which is not supported by NSString. Both NSString and NSPredicate allow the %@ variable for substituting Objective-C objects into the string. NSPredicate does not support the fprint-style format variables such as %d, %f, or %s. It does however, add a new substitution variable not used by NSString: %K.

The %K substitution variable is used for key paths and attribute names. If the name of the attribute you want to compare might change, you cannot use %@ to do so. So, for example, this won't work:

```
NSPredicate *pred = [NSPredicate predicateWithFormat:
  @"%@ == %@", @"name", @"Joe"];
```

The reason that this doesn't work is that NSPredicate recognizes the %@ operator only for substituting *values* not for substituting *keys*. When you are specifying an attribute name or key path, you have to use %K instead of %@. To correct that previous code example, we simply substitute %K for the first %@ like so:

```
NSPredicate *pred = [NSPredicate predicateWithFormat:
  @"%K == %@", @"name", @"Joe"];
```

Fetching with Predicates

Now that you've been introduced to predicates, we're ready to write the rest of our Core Data utility functions. First, let's write a utility method for a very common fetch: fetching all entities where one particular attribute has a particular value. Although `NSPredicate` allows very complex queries, you're likely to find that the bulk of the queries you actually use in your applications are relatively simple, and this method will make life easier in those situations.

MTCDUtilities.m – objects objectsForEntityNamed:matchingKey:andValue:inContext

```
+(NSArray *) objectsForEntityNamed:(NSString *)name
  matchingKey:(NSString *)key
  andValue:(id)value
  inContext:(NSManagedObjectContext *)context
{
  // Since NSString and NSPredicate use different format strings,
  // we use a two-step process to create our format string here

  NSString *predString = [NSString stringWithFormat:
    @"%@ == %%@", key];
  NSPredicate *pred = [NSPredicate
    predicateWithFormat:predString, value];

  // We still need an entity description, of course

  NSEntityDescription *entity = [NSEntityDescription
    entityForName:name inManagedObjectContext:context];

  // And, of course, a fetch request. This time we give it both the entity
  // description and the predicate we've just created.

  NSFetchRequest *req = [[NSFetchRequest alloc] init];
  [req setEntity:entity];
  [req setPredicate:pred];

  // We declare an NSError and handle errors by raising an exception,
  // just like in the previous method

NSError *error = nil;
  NSArray *array = [context executeFetchRequest:req
    error:&error];
  if (array == nil)
  {
    NSException *exception = [NSException
      exceptionWithName:MTCoreDataExeption
      reason:[error localizedDescription]
      userInfo:nil];
    [exception raise];
  }

  // Now, release the fetch request and return the array

  [req release];
  return array;
}
```

Very handy. Now, we have a convenience method for retrieving object instances matching a single key/value pair, like so:

```
NSArray *results = [MTCDUtilities objectsForEntityNamed:
  @"person" matchingKey:@"lastName"
  andValue:@"Smith"
  inContext:context];
```

More importantly, you now know the basics of creating a fetch request using a predicate, so should be able to craft more complex fetch requests for situations where this method is inadequate.

Now, in practice, you'll likely find yourself using the method above a lot of times with unique identifiers. In those situations, you know you'll only retrieve a single object, but yet you'll still get back an array. For situations where you know there will only be a single matching object, perhaps another convenience method is in order.

MTCDUtilities.m – objectForEntityNamed:matchingKey:andValue:inContext:

This method returns a single object matching a single key/value pair. If more than one object is actually returned, an exception is thrown.

```
+(NSManagedObject *)objectForEntityNamed:(NSString *)name
  matchingKey:(NSString *)key
  andValue:(id)value
  inContext:(NSManagedObjectContext *)context
{
```

// We call the previous method, then return the object at index 0. We
// declare no exception handler, so an exception encountered in this call
// will stop execution of this method and throw up to the code
// from where it was called.

```
  NSArray *array = [MTCDUtilities
    objectsForEntityNamed:name
    matchingKey:key andValue:value inContext:context];
```

// If there are more than one objects in the array, throw an exception

```
  if ([array count] > 1)
  {
    NSException *exception = [NSException
exceptionWithName:MTTooManyEntitiesReturnedException
      reason:@"Too many instances retrieved for criteria"
      userInfo:nil];
    [exception raise];
  }
```

// If there are no objects, just return nil

```
if ([array count] == 0)
    return nil;
```

// Return the object at index 0

```
return (NSManagedObject *)[array objectAtIndex:0];
}
```

Now you'll be able to pull back a specific item with aplomb:

```
NSManagedObject *item = [MTCDUtilities
  objectsForEntityNamed:@"book" matchingKey:@"id"
  andValue:[NSNumber numberWithInt:192]
inContext:context];
```

Notice that I passed an `NSNumber` instead of a string? That's not only acceptable, it's the correct thing to do when the attribute you're using is a numeric value, just as you would pass an `NSDate` if you were comparing a date attribute.

Compounding Predicates

But wait! There's more! If you order in the next ten minutes, I'll throw in this free set of Ginsu™ knives. Okay, not really, but I will throw in one more handy method. You won't always be able to get away with using queries based on a single key-value pair. Sometimes you'll need, for example, to pull back a person based on a first AND a last name, or pull back all entities of one type OR another type. These situations are accomplished with a specialized subclass of `NSPredicate` called `NSCompoundPredicate`.

Given any two or more existing `NSPredicates`, you can create a compound predicate using the logical operators `AND`, `OR`, or `NOT`. You simply create an array with all the predicates you want to join, and pass them into one of `NSCompoundPredicate`'s convenience class methods, like so:

```
NSArray *preds = [NSArray arrayWithObjects:
  pred1,pred2, nil];
NSPredicate *notPred = [NSCompoundPredicate
  notPredicateWithSubpredicates:preds];
NSPredicate *andPred = [NSCompoundPredicate
  andPredicateWithSubpredicates:preds];
NSPredicate *orPred = [NSCompoundPredicate
  orPredicateWithSubpredicates:preds];
```

You can even compound existing compound predicates, which we'll do in this last method.

MTCDUtilities.m – objectsForEntityNamed:matchingKeysAndValues:usingOR: inContext

Finds all instances of a given entity based on an `NSDictionary` of key/value pairs. The key-value pairs will be turned into equality predicates and compounded using either `OR` or `AND` depending on the value of `useOR`.

```
+(NSArray *)objectsForEntityNamed:(NSString *)name
  matchingKeysAndValues:(NSDictionary *)keyValues
  usingOR:(BOOL)useOR
  inContext:(NSManagedObjectContext *)context
{
```

// We'll retrieve an enumerator of all the keys in the dictionary
NSEnumerator *e = [keyValues keyEnumerator];

// Declare the predicate outside of the enumerator loop

```
NSPredicate *pred = nil;
```

// Declare a string to hold the current key while looping

```
  NSString *key;

  while (key = [e nextObject])
  {
```

// Declare a format string for creating the current subpredicate

```
    NSString *predString =
      [NSString stringWithFormat:@"%@ == %%@",  key];
```

// First time through, `pred` is `nil` and shouldn't be compounded with anything

```
    if (pred == nil)
      pred = [NSPredicate predicateWithFormat:predString,
        [keyValues objectForKey:key]];
    else
    {
```

// if `pred` is not nil, then create a compound based on the new
// subpredicate `tempPred` and the existing predicate `pred`

```
      NSPredicate *tempPred = [NSPredicate
        predicateWithFormat:predString,
        [keyValues objectForKey:key]];
      NSArray *array = [NSArray
arrayWithObjects:tempPred,
        pred, nil];
      if (useOR)
      pred = [NSCompoundPredicate
        orPredicateWithSubpredicates:array];
    else
      pred = [NSCompoundPredicate a
        ndPredicateWithSubpredicates:array];
  }
}

// Everything from here down should look familiar.

NSEntityDescription *entity = [NSEntityDescription
  entityForName:name inManagedObjectContext:context];
NSFetchRequest *req = [[NSFetchRequest alloc] init];
[req setEntity:entity];
[req setPredicate:pred];
NSError *error = nil;
NSArray *array = [context executeFetchRequest:req
  error:&error];
if (array == nil)
{
  NSException *exception = [NSException
    exceptionWithName:MTCoreDataExeption
    reason:[error localizedDescription]
    userInfo:nil];
  [exception raise];
}
[req release];
return array;
}
```

To use this last method, simply pack a dictionary with the attributes as the keys and the values to compare them to as the values, like so:

```
NSDictionary *dict = [NSDictionary
  dictionaryWithObjectsAndKeys:@"BookCo", @"publisher",
  [NSNumber numberWithInt:482769], @"salesRank", nil];
NSArray *array = [MTCDUtilities objectsForEntityNamed:
  @"Book" matchingKeysAndValues:dict usingOR:YES
  inContext:[self managedObjectContext]];
```

Conclusion

This article gives you the last missing major building block for creating Core Data applications. Between this article and its two predecessors, you should now feel comfortable creating a Core Data data model, hooking it into your interface, interacting with it programmatically, and finding data within your application's context.

Core Data may seem a little intimidating at first, especially if you've never worked with an object-relational mapping tool such as EOF or Cayenne before. Hopefully these three articles have given you enough information to realize that Core Data really isn't scary at all, but that it can give you a frightening increases in your productivity.

MT

About The Author

Jeff LaMarche wrote his first line of code in Applesoft Basic on a Bell & Howell Apple //e in 1980 and he's owned at least one Apple computer at all times since. Though he currently makes his living consulting in the Mac-unfriendly world of "Enterprise" software, his Macs remain his first and greatest computer love. You can reach him at jeff_lamarche@mac.com.

Long Distance

2.9¢ Per Minute!

Straight 6 second billing increments

Excellent rates on intrastate, intralata/toll calls and international calling with no term contract.

Toll Free (800/888/877/866) service, same low per minute rate for incoming calls.

10 cents per minute calling card.

Detailed billing directly from OPEX.

Quality electronic and telephone customer support.
No monthly billing fee if your bill is over $20.00 each month.

(NOTE: $2.00 billing fee is charged when your bill is under $20.00.)

www.lowcostdialing.com

BACK TO BASH BASICS: PART 2

TIME TO ADVANCE OURSELVES

With any exercise, you need to continually push yourself. Without that extra effort, what once was a challenge becomes easy, something to drift through. At the same time, you may be missing advanced techniques that make other areas easier or more efficient. Similarly, shell scripting can go many layers deep, and you can exercise your knowledge in many ways. Last month in, "Back to bash Basics Part 1," we focused on flow control. You may have noticed some of the things I didn't cover explicitly. There's always more to learn! Let's tie up those loose ends.

More Looping

Since we discussed looping constructs so much last month, that's where we'll pick up. In the `select` example, you'll see a `break` statement – that could use some explanation. `break` simply terminates the current loop. If it were removed from the `select` example, you would be asked repeatedly which file you want to inspect. Let's see what that would look like:

```
#!/bin/bash

select theItem; do
        if [ $theItem ]; then
                file $theItem
        fi
done
```

When this is run, the output looks like this:

```
Jack-Kerouak:~/bin marczak$ ./st.sh *
1) BidToJob.dmg
2) bomcheck.sh
3) cl.txt
4) createdmg
5) diskrep.sh
6) exscript
#? 3
cl.txt: ASCII text
#? 4
createdmg: ASCII text
#? 5
diskrep.sh: Bourne-Again shell script text executable
#? ^C
```

Notice that this time, we need to press ctrl-c to stop the program. `break` applies to any loop:

```
#!/bin/bash
for i in $*;
        do
        if [ ! -O $i ]; then
                echo You do not own $i!  I am outta here!
                break
        fi
        echo $i is your file!
done
```

Running as me, I'm shown:

```
$ ./bt.sh *
BidToJob.dmg is your file!
bomcheck.sh is your file!
bt.sh is your file!
cl.txt is your file!
[…clipped for brevity]
```

Running in that same directory as root gives us:

```
# ./bt.sh *
You do not own BidToJob.dmg! I am outta here!
```

AOT (or, how the shell will separate files)

Have you ever crossed your AOT (Acronym Overload Threshold)? "There's a problem with the RIP!" Raster Image Processor, or Routing Information Protocol? While there's only so many TLAs (Three Letter Acronyms) that you can deal with, I need you add one more: IFS (no, not Iterative Fractal Systems!). The shell uses the **I**nternal **F**ield **S**eparator to determine how to break apart tokens, and how to separate incoming parameters. By default, IFS is equal to space, tab and newline.

When we discussed the `for` loop last month, several things were quickly touched upon that can be expanded. In addition to the `$@` variable, which expands to individual double-quoted strings, there is the `$*` variable, which is a single string containing each positional parameter. How do you know where each parameter breaks? `$*` separates each parameter by using the first character of your IFS variable. We'll get back to how this can be very useful.

Also, last month showed an example that looked something like this:

```
FILES=`ls *.sh`

for i in $FILES
do
        ...
done
```

This example 'just works' because `ls *.sh` will separate its output with linefeeds. Hey, that's one of the characters in IFS by default! What good fortune! This same example will fall apart if you reassign the IFS variable prior to the loop:

```
IFS="-"
FILES=`ls *.sh`
for i in $FILES; do
...
done
```

$i will still hold $FILES, but it won't be tokenized – not the way you'd expect (the line feeds will still be in there, but $i won't break on them).

So, then, why would we ever touch IFS? Well, what if you wanted to search through something that is ***not*** broken up by a space, newline or tab? Like $PATH, for instance:

```
#!/bin/bash

IFS=:

for theDir in $PATH
do
        theLatest=`ls -lotr $theDir | tail -1`
        echo Newest file in $theDir:
        echo $theLatest
        echo
done
```

This simply goes through our $PATH and tells us the newest file in each directory. Could be useful.

Oh, and another thing

Like Apple, shell scripting always seems to have "one more thing." For this month, this thing comes in the form of being able to effectively handle parameters. Time to introduce `shift` and `getopts`.

When writing a script, parameters can be accessed a few ways. If you always rely on direct access ($1, $2...etc), you run into some limits. One way to simply loop through all parameters is to use `shift`. `shift` makes $1 = $2, $2 = $3...etc. You lose the first value that was assigned to $1. To look for a few specific parameters, you can loop through the values:

```
#!/bin/bash

while [ `echo $1 | grep "-"` ]; do
        case $1 in
                -a ) echo "You supplied the -a flag";;
                -b ) echo "You supplied the -b flag";;
                -c ) echo "You supplied the -c flag";;
                * ) echo "Usage: $0 -a -b -c";
                exit 1;;
        esac
        shift
done
```

Run this code and you'll see:

```
$ ./shifttest.sh -b -a
You supplied the -b flag
You supplied the -a flag
```

Now, shift is cool, and still comes in handy, but to truly handle command-line options smoothly, we have to employ `getopts`. Sure, you can roll your own each time, however, people have come to expect their options to behave in certain ways. One should be able to combine options, as with `tar`, for example: `tar -xzvf blah.tar.gz`. Basically, you don't need to roll your own because `getopts` exists.

`getopts` allows you to handle options in a standard way. Seeing it in action is the quickest way to get up to speed:

```
#!/bin/bash

while getopts ":xyz:t" theOption; do
        case $theOption in
                x ) echo "Option x chosen";;
                y ) echo "Option $theOption chosen";;
                z ) echo "Option z chosen with argument:
$OPTARG";;
                t ) echo "Option t chosen";;
                \? ) echo "Unknown option chosen"
                        exit 1;;
                * ) echo "You need to supply an option!"
                        exit 2;;
        esac
done
```

getopts is designed to be dumped in a loop that will feed it arguments passed into the script. It accepts a string that defines the allowed options, followed by a variable that will hold the current option, sans the "-" or "+" (nicely, either are allowed). Using getopts will define two variables: $OPTIND, the current index and $OPTARG, the current argument passed with an option. Running this produces this output:

```
$ ./gotest.sh -yxz test
Option y chosen
Option x chosen
Option z chosen with argument: test
```

The string that getopts accepts can only contain letters and the colon character. Each letter is an option you wish to support. If a letter is followed by a colon, that tells getopts that an argument is required. By having a lead colon character in the parameter list string, you suppress the error message that getopts will print if an option is not recognized. In either case, an unrecognized option will set the variable to "?", so you can deal with it.

Put it all Together

I've gotten a request or two asking how to deal with math in the shell. While there are specialized CLI apps that will deal with arithmetic, the shell can do some basic functions, and sometimes, that's all you need. The trick is the underused declare statement. declare tells the shell how you want to treat variables, which are strings by default. So, this doesn't do what one would expect:

```
$ number1=7
$ number2=8
$ total=number1*number2
```

When you echo $total, you get "number1*number2": strings. We need to tell the shell that $total should be treated as an integer:

```
$ declare -i total
$ total=number1*number2
$ echo $total
56
```

Much better! declare can define several different types of variables:

-a	variable is an array
-i	treat as integer
-r	makes variable read-only
-x	automatic export (like the 'export' built-in)

There are some others, but this is all we need concentrate on for now. All of the usual suspects are available as mathematic operators:

+	Addition
-	Subtraction
*	Multiply
/	Divide
%	Remainder
<<	Bit-shift left
>>	Bit-shift right
&	Bitwise and
|	Bitwise or
~	Bitwise not
!	Bitwise not
^	Xor

In addition to declare-ing a variable to be an integer, you can use let to make the assignment:

```
let theTotal='5 * 7'
```

Ah, let....brings me back to my C64 BASIC days...

Now, you should be able to write fairly sophisticated shell script that includes slick input processing, good error handling and even some basic computations!

Make Yourself Useful...

...to everyone. Just remember that bash scripting will help you not only with OS X, but with Linux, IRIX, FreeBSD, and even Windows – if you install a Unix shell there (which can be had for free from Cygwin or Microsoft).

This month highlights the fact that shell scripting is relatively ***easy***, can be ***fun*** and ***powerful***. Even better, you'll find bash built-in to every OS X machine you touch! Let this all sink in: while I'll get back to bash scripting in future columns, more Unix detours next month!

MT

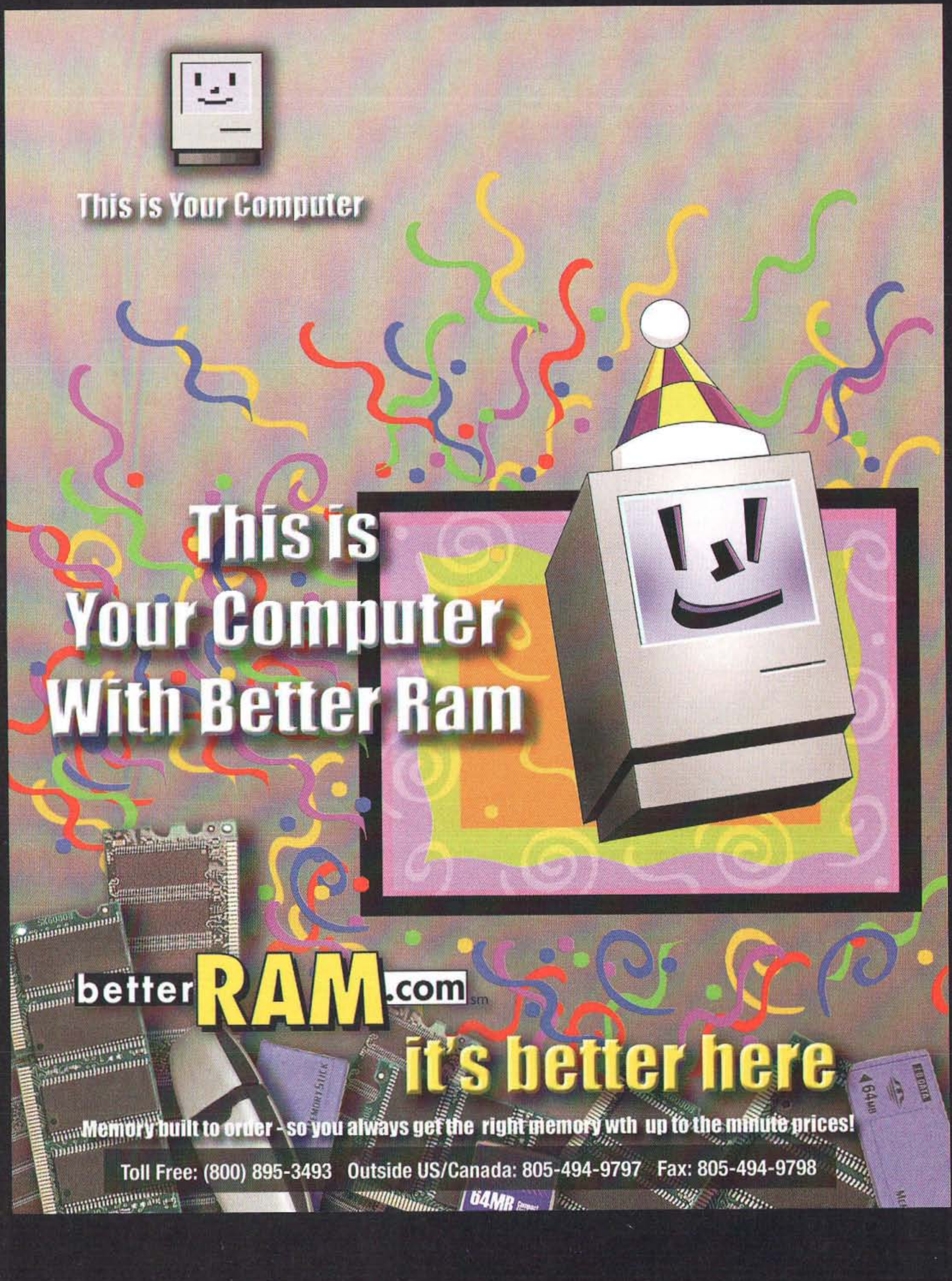
About The Author

Ed Marczak keeps it simple. Tech simplicity at http://www.radiotope.com

MACTECH®

Get MacTech delivered to your door at a price FAR BELOW the newstand price. And, it's RISK FREE!

store.mactech.com/riskfree

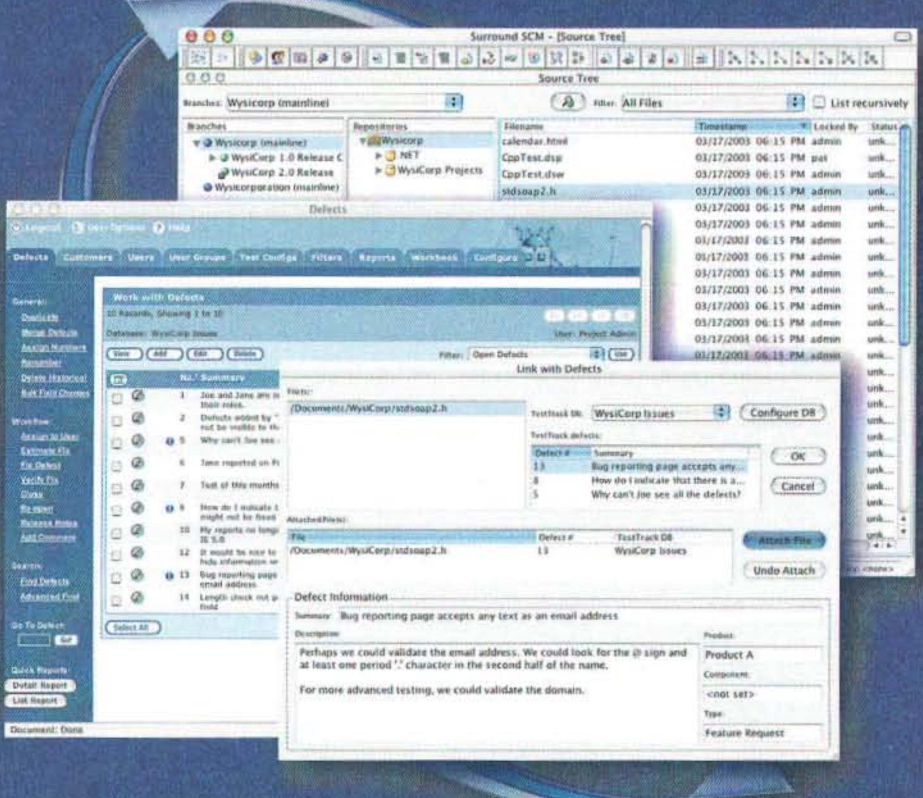

This is Your Computer
This is Your Computer With Better Ram
betterRAM.com sm
it's better here
Memory built to order - so you always get the right memory wth up to the minute prices!
Toll Free: (800) 895-3493 Outside US/Canada: 805-494-9797 Fax: 805-494-9798

Advertiser/Product Index

The index on this page is provided as a service to our readers. The publisher does not assume any liability for errors or omissions.

Complete Source Control and Defect Management for Mac OS X

Seapine Software™
changing the world of software development


Surround SCM - [Source Tree]

Source Tree

Branches: Wysicorp (mainline) — Filter: All Files — List recursively

Filename	Timestamp	Locked By	Status
calendar.html	03/17/2003 06:15 PM	admin	unk...
CppTest.dsp	03/17/2003 06:15 PM	pat	unk...
CppTest.dsw	03/17/2003 06:15 PM	admin	unk...
stdsoap2.h	03/17/2003 06:15 PM	admin	unk...

Link with Defects

Files: /Documents/WysiCorp/stdsoap2.h

TestTrack DB: WysiCorp Issues

TestTrack defects:

Defect #	Summary
13	Bug reporting page accepts any...
8	How do I indicate that there is a...
5	Why can't Joe see all the defects?

Attached File(s):

File	Defect #	TestTrack DB
/Documents/WysiCorp/stdsoap2.h	13	WysiCorp Issues

Defect Information

Summary: Bug reporting page accepts any text as an email address

Description: Perhaps we could validate the email address. We could look for the @ sign and at least one period '.' character in the second half of the name.

For more advanced testing, we could validate the domain.

Product: Product A

Component: <not set>

Type: Feature Request

Effective source code control and defect tracking require powerful, flexible, and easy-to-use tools—Surround SCM and TestTrack Pro

- Complete source code control with private workspaces, automatic merging, role-based security, and more
- Comprehensive defect management — track bug reports and change requests, define workflow, customize fields
- Fast and secure remote access to your source files and defects — work from anywhere
- Advanced branching simplifies managing multiple versions of your products
- Link code changes with defects and change requests — know who changed what, when, and why
- Scalable and reliable cross-platform, client/server solutions support Mac OS X, Windows, Linux, and Solaris
- Exchange data using XML and ODBC, extend and automate with SOAP support
- Licenses priced to fit your budget

Seapine Software Product Lifecycle Management
Award winning, easy-to-use software development tools

Seapine **Surround SCM**

Seapine **TestTrack PRO**

software development 13th annual product excellence award JOLT

Download Surround SCM and TestTrack Pro at www.seapine.com or call 1-888-683-6456

Welcome to the new Tips & Tidbits!

This new column is your opportunity to spread the word about little bits of information that you know or find out about. These tidbits can address any type of useful information: from user tips to networking to admin to programming.

When the tips you submit are printed in the magazine, MacTech will send you a coveted t-shirt, just for MacTech tipsters. This is the only way to get these shirts, and show that you know your stuff.

To submit a tip, go to the MacTech web site at **http://www.mactech.com/tips**

Spread the word about things you know that others could benefit from knowing!

Making The Invisible Files Visible

Are you on a Mac OS X Server machine, and you want to be able to see all the files all the time? Maybe your the admin for the machine, and you are sick of having to bring up the Find window and looking for the invisible files that way.

There is a setting in the Finder settings that will allow you to show all the files. To do this, simply pull up your Terminal app on the machine in question, type in:

```
defaults write com.apple.finder AppleShowAllFiles YES
```

press return, and then either log in/out, or restart the computer. After that, all files on that machine will be visible.

To undo this, do the reverse.

```
defaults write com.apple.finder AppleShowAllFiles NO
```

Neil Ticktin

Mac OS X Server Admin AFP Access

Are you looking to set up AFP access for the administrator on your Mac OS X Server machine that allows you to see the entire drive on the machine running Mac OS X Server?

The default setup for a user allows you to see three subvolumes — Groups, Users, and Public. If you want to administer the web sites folders, or even things at the root level of the drive, you have to give yourself additional privledges.

The easiest way to do this for the administrator is through the Terminal app. Just type in:

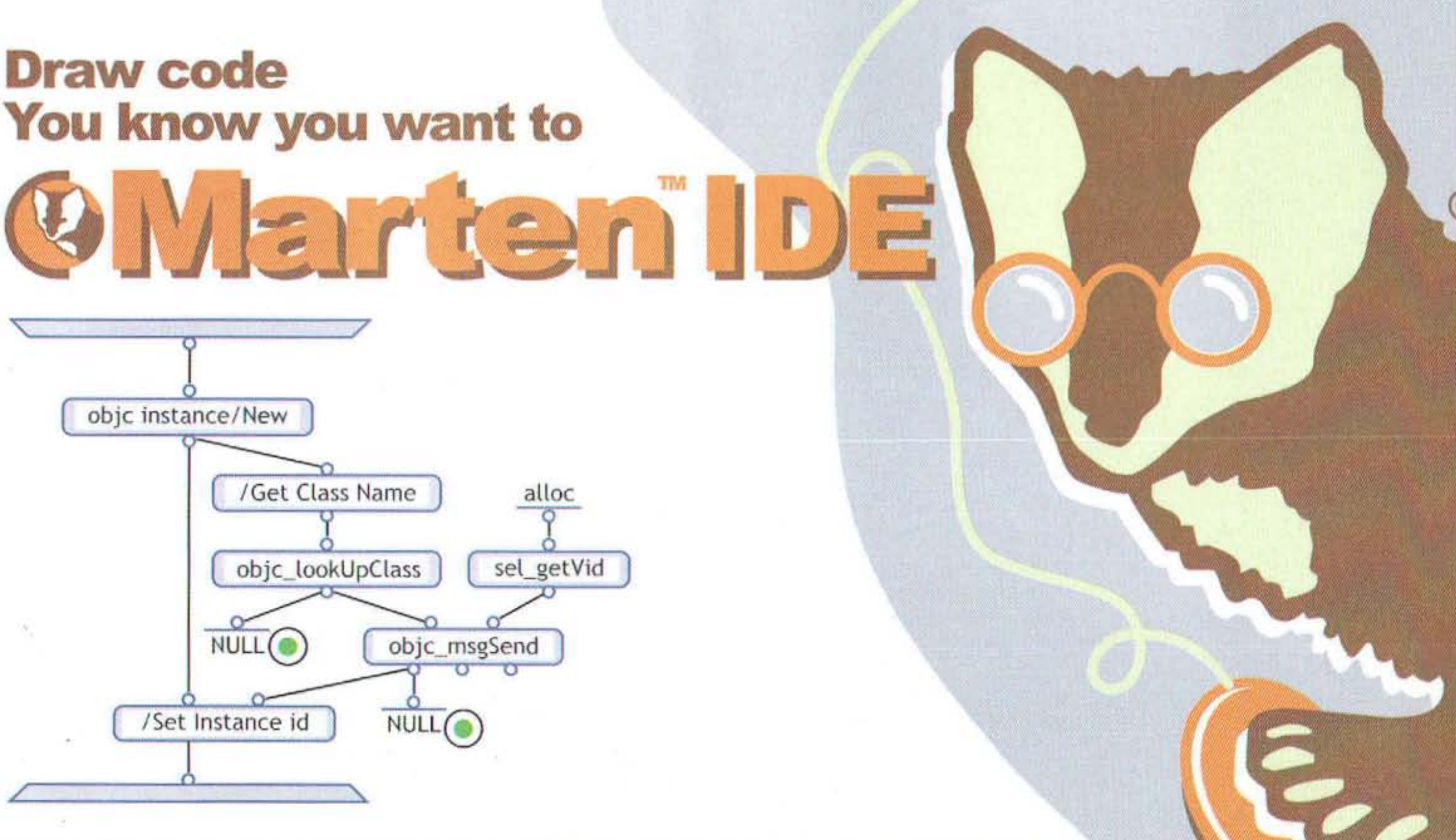
```
sudo serveradmin settings afp:admin31GetsSp = no
```

Thanks to Dean Shavit at Macworkshops.com for the background on this.

Neil Ticktin

Draw code
You know you want to

Marten™ IDE



• Edit while you run • One click Cocoa and Carbon application creation • Powerful debugger

w w w . a n d e s c o t i a . c o m

Multiple Formats. Multiple Platforms. Complex Installers.

We have the solution:

StuffIt Engine SDK

Solving the compression & multi-platform puzzle!
www.stuffit.com/sdk/

Licenses start as low as $99/Yr.

Put the power of StuffIt to work for you.

- Adds value to your applications by integrating compression and encryption tools.
- The only tool that supports the StuffIt file format.
- Make self extracting archives for Macintosh or Windows
- Available for Macintosh, Windows, Linux or Solaris

Mac

solaris

StuffIt InstallerMaker

An OS X native version ready for developers!
www.stuffit.com/installermaker/

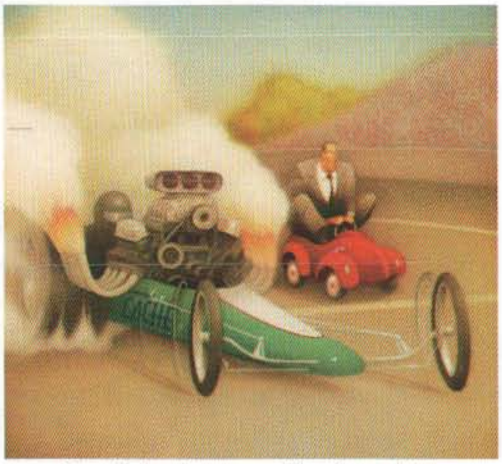
Prices start at $250

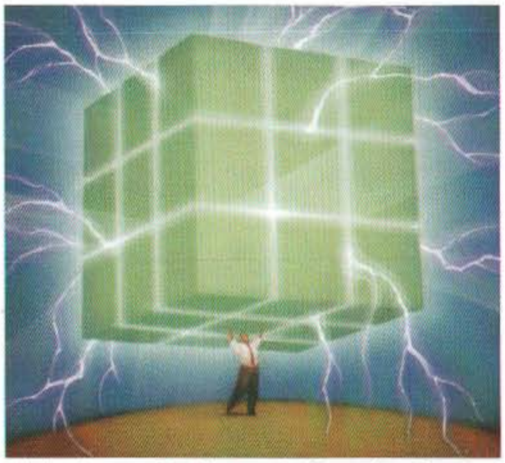
Give your software a solid beginning

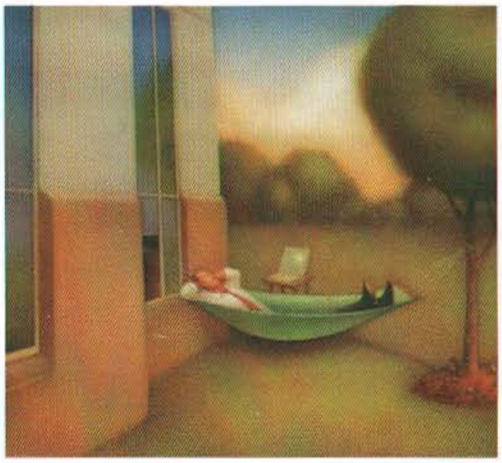
- Create Macintosh OS X and Macintosh Classic compatible installers
- Get all the tools you need to install, uninstall or update your software in one complete package
- Add muscle to your installers by customizing your electric registration form to include surveys and special offers.

Allume Systems
www.allume.com
email: dev.sales@allume.com

Innovations by InterSystems


Rapid development with robust objects


Lightning speed with a multidimensional engine


Easy database administration

Massive scalability on minimal hardware

Database For Web-Based Applications.

Caché, the multidimensional database from InterSystems, can automatically project data and logic in a number of Web-centric forms, such as XML, Web Services, Java, and EJB. These unique capabilities make Caché ideal for rapidly developing Web-based applications.

Caché is the first database to seamlessly combine robust objects and robust SQL, thus eliminating object-relational mapping. Its post-relational technology delivers lightning-fast transaction processing, real-time analytics, and massive scalability on minimal hardware. It requires little administration, and incorporates a rapid application development environment.

These innovations mean faster time-to-market, lower cost of operations, and higher application performance. We back these claims with this money-back guarantee: *Buy Caché for new application development, and for up to one year you can return the license for a full refund if you are unhappy for any reason.** Caché is available for Unix, Linux, Windows, Mac OS X, and OpenVMS – and it's deployed on more than 100,000 systems ranging from two to over 50,000 users. We are InterSystems, a global software company with a track record of innovation for more than 25 years.

InterSystems CACHÉ™

Try an innovative database for free: Download a fully functional, non-expiring copy of Caché, or request it on CD, at **www.InterSystems.com/Cache11HH**

* Read about our money-back guarantee at the web page shown above.